

Titulació:

GRAU EN ENGINYERIA ELECTRÒNICA INDUSTRIAL I AUTOMÀTICA

Alumne (nom i cognoms):

ÁLVARO NOVALES FELICES

Enunciat TFG/TFM:

INTERCONEXIÓN DE EQUIPOS INDUSTRIALES CON MODBUS PARA
AUTOMATIZACIÓN DE PROCESOS

Director/a del TFG/TFM:

DAVID ROMERO DURAN

Convocatòria de lliurament del TFG/TFM:

MAIG 2020

ABSTRACT

Castellano

Este proyecto diseña un sistema de control y supervisión de un motor mediante la interconexión de equipos industriales. Ambas funciones, se pueden realizar mediante dos fuentes seleccionables que actúan sobre un convertidor de frecuencia. Una fuente es una maqueta que actúa en las entradas y salidas analógicas y digitales y la otra un autómata programable y un panel HMI que actúan sobre la memoria del autómata mediante un bus de campo, Modbus RTU.

El control permite actuar sobre el motor a diferentes velocidades y permite la parametrización del convertidor. Asimismo, el sistema de supervisión permite visualizar a tiempo real el estado del sistema y los parámetros más significativos del convertidor.

English

This project designs a motor supervision and control system through the interconnection of industrial equipment. Both functions, can be carried out by means of two selectable sources that act on a frequency converter. One source is a panel that acts on the analogic and digital inputs and outputs and the other source is a programmable controller and an HMI panel that act on the memory using a fieldbus, Modbus RTU protocol.

The control allows acting on the motor at different speeds and the parameterization of the converter. The supervision system allows to visualize at real-time the state of the system and the most significant parameters of the frequency converter.

AGRADECIMIENTOS

Con estas líneas quiero expresar mi más sincero agradecimiento a todas aquellas personas que han contribuido en la realización de este proyecto.

En primer lugar a mi tutor del proyecto David Romero Duran, por su apoyo técnico, consejo y disposición a lo largo de este periodo.

A mi familia y en especial a mis padres, sin su apoyo y esfuerzo nada de todo esto hubiera sido posible.

A Josep Ballesteros y a Josep Canudas, por su ayuda y colaboración.

A todos, muchas gracias.

ÍNDICE

1. INTRODUCCIÓN	10
2. OBJETO Y ALCANCE DEL PROYECTO	13
3. METODOLOGÍA	14
3.1. Desarrollo	14
3.2. Diagrama de Gantt	14
4. DESCRIPCIÓN DEL SISTEMA	16
5. EL CONVERTIDOR	17
5.1. Fundamentos teóricos	17
5.2. Convertidor ACS480	20
5.2.1. Características técnicas	22
5.2.2. Macros y configuración de parámetros	26
5.3. Puesta en marcha del convertidor	30
6. EL AUTÓMATA PROGRAMABLE	31
6.1. Fundamentos teóricos	31
6.2. Autómata programable AC500	35
6.2.1. Terminal base TB511	36
6.2.2. PAC PM573 - ETH	37
6.2.3. Módulo DA501	39
6.2.4. Módulo CD522	40
7. HMI	43
7.1. Fundamentos teóricos	43
7.2. Panel CP630	44
8. MAQUETA	46
9. PROTOCOLOS DE COMUNICACIÓN INDUSTRIAL	46
9.1. Modbus	47
9.1.1. Modbus RTU	48
9.1.2. Modbus TCP	48

10. HERRAMIENTAS SOFTWARE.....	49
10.1. Drive composer	49
10.2. Automation Builder	49
10.2.1. Codesys	50
10.3. Panel Builder	50
11. COMUNICACIÓN	51
11.1. Comunicación PC-PAC	51
11.1.1. Configuración PC	51
11.1.2. Configuración PAC	53
11.2. Comunicación PAC-Convertidor.....	54
11.2.1. Configuración PAC	55
11.2.2. Configuración convertidor	56
11.3. Comunicación PAC-HMI Modbus RS-485	57
11.3.1. Configuración PAC	57
11.3.2. Configuración HMI	57
11.4. Comunicación PAC-HMI Modbus TCP	58
11.4.1. Configuración PAC	59
11.4.2. Configuración HMI	60
12. CONTROL Y SUPERVISIÓN DEL MOTOR.....	61
12.1. Programación del convertidor.....	61
12.2. Programación del autómatas	61
12.2.1. Programa PLC_PRG	62
12.2.2. Programa ACS480	62
12.2.2.1. Bloque de funciones ACS_COM_MOD_RTU.....	62
12.2.2.2. Bloque de funciones ACS_DRIVES_CTRL_STANDARD	63
12.2.2.3. Bloque de funciones ACS_MOD_READ_N_PRM.....	63
12.2.2.4. Bloque de funciones ACS_MOD_WRITE_N_PRM	63
12.2.2.5. Código programa ACS480.....	64
12.2.3. Programa MARCHA_PARO	64
12.2.4. Programa EJECUTAR_L_E.....	64

12.2.5.	VEL_SEL_FB	64
12.2.6.	AO2_SEL_PRG	65
12.2.7.	AJUSTES_VAR.....	65
12.3.	Programación HMI.....	66
12.4.	Funcionalidad de la Maqueta	69
12.5.	Motor seleccionado para las pruebas	71
13.	CONCLUSIONES	72
14.	BIBLIOGRAFIA	74
15.	ANEXOS	76
15.1.	Introducción a las herramientas software.....	76
15.1.1.	Automation Builder.....	76
15.1.1.1.	Descarga e instalación.....	76
15.1.1.2.	Descripción del entorno	76
15.1.1.3.	Creación de un proyecto.....	78
15.1.1.4.	Configuración del autómeta	80
15.1.1.5.	Bibliotecas.....	81
15.1.2.	Codesys	83
15.1.2.1.	Descarga e instalación.....	83
15.1.2.2.	Descripción del entorno	83
15.1.2.3.	Declaración de variables.....	84
15.1.2.4.	Exportar variables	85
15.1.2.5.	Lenguajes de programación	85
15.1.3.	Panel Builder 600.....	86
15.1.3.1.	Descarga e instalación.....	86
15.1.3.2.	Descripción del entorno	87
15.1.3.3.	Creación de un proyecto.....	88
15.1.3.4.	Importar variables	88
15.2.	Control y supervisión del motor	90
15.2.1.	Programación del convertidor	90
15.2.2.	Programación PAC	98

15.2.2.1. Programa ACS480	98
15.2.2.1.1. Bloque de funciones ACS_COM_MOD_RTU.....	98
15.2.2.1.2. Bloque de funciones ACS_DRIVES_CTRL_STANDARD	100
15.2.2.1.3. Bloque de funciones ACS_MOD_READ_N_PRM.....	101
15.2.2.1.4. Bloque de funciones ACS_MOD_WRITE_N_PRM	103
15.2.2.2. Código programa ACS480.....	104

ÍNDICE DE FIGURAS

Figura 1: Planificación inicial	15
Figura 2: Arquitectura del sistema.....	16
Figura 3: Esquema convertidor de frecuencia indirecto	17
Figura 4: Forma de la tensión después de la etapa de rectificado	18
Figura 5: Señal de control triangular comparada con la señal de referencia.....	19
Figura 6: Variador de frecuencia ACS480.....	20
Figura 7: Descripción general del hardware.....	20
Figura 8: Esquema de las conexiones de control.	21
Figura 9: Diagrama simplificado del circuito de potencia del ACS480.....	22
Figura 10: Estructura básica de un autómata programable	31
Figura 11: Mapa de memoria de un PLC	33
Figura 12: Autómata programable de la gama AC500.....	35
Figura 13: Terminal base TB511	36
Figura 14: PM573-ETH.....	37
Figura 15: Funcionalidades e indicaciones físicas PM573-ETH	37
Figura 16: Módulo de expansión DA501	39
Figura 17: Funcionalidades e indicaciones físicas DA501	39
Figura 18: Módulo de expansión CD522.....	41
Figura 19: Funcionalidades e indicaciones físicas CD522.....	41
Figura 20: CP630	44
Figura 21: Dimensiones y puertos del panel CP630	44
Figura 22: Maqueta	46
Figura 23: Configuración de la IP del PC	52
Figura 24: Configuración de la IP del PC.	52
Figura 25: Configuración de la IP del PC	53
Figura 26: Configuración de la IP del PAC.....	53
Figura 27: Configuración de la IP del PAC.....	54
Figura 28: Configuración de la IP del PAC.....	54

Figura 29: Configuración en RS-485 del PAC.....	55
Figura 30: Configuración en RS-485 del PAC.....	55
Figura 31: Configuración en RS-485 del PAC.....	57
Figura 32: Configuración en RS-485 del HMI	58
Figura 33: Configuración en RS-485 del HMI	58
Figura 34: Configuración en TCP/IP del PAC	59
Figura 35: Configuración en TCP/IP del PAC	59
Figura 36: Configuración en TCP/IP del HMI	60
Figura 37: Configuración en TCP/IP del HMI	60
Figura 38: Programa PLC_PRG.....	62
Figura 39: Bloque ACS_COM_MOD_RTU	62
Figura 40: Bloque ACS_DRIVES_CTRL_STANDARD	63
Figura 41: Bloque MOD_READ_N_PRM	63
Figura 42: Bloque MOD_WRITE_N_PRM	63
Figura 43: Programa MARCHA_PARO.....	64
Figura 44: Programa EJECUTAR_L_E	64
Figura 45: Programa VEL_SEL_FB	64
Figura 46: Programa AO2_SEL_PRG.....	65
Figura 47: Programa AJUSTES_VAR.....	65
Figura 48: Pantalla de Inicio	66
Figura 49: Pantalla de Parámetros.....	66
Figura 50: Pantallas Gráficos	67
Figura 51: Cuadro diálogo Configuración.....	68
Figura 52: Pantallas Configuración	68
Figura 53: Pantalla cambio usuario	69
Figura 54: Motor ABB 4kW.....	71
Figura 55: Evolución del proyecto	73
Figura 56: Entorno de Automation Builder	76
Figura 57: Crear un proyecto nuevo Automation Builder	78
Figura 58: Selección autómeta.....	79

Figura 59: Agregar módulos de extensión al autómeta	80
Figura 60: Configuración PM573-ETH	81
Figura 61: Hardware PM573-ETH.....	81
Figura 62: Adición bibliotecas Codesys	82
Figura 63: Entorno de Codesys.....	83
Figura 64: Declaración variables Codesys.....	85
Figura 65: Entorno Automation Builder	87
Figura 66: Crear un proyecto nuevo Panel Builder 600	88
Figura 67: Importación variables Panel Builder.....	89
Figura 68: Declaración variables Panel Builder	90
Figura 69: Bloque ACS_COM_MOD_RTU	98
Figura 70: Bloque ACS_DRIVES_CTRL_STANDARD.....	100
Figura 71: Bloque MOD_READ_N_PRM	101
Figura 72: Bloque MOD_WRITE_N_PRM	103
Figura 73: Programa ACS480	105

ÍNDICE DE TABLAS

Tabla 1: Equipos y software	16
Tabla 2: Etiquetas del hardware	21
Tabla 3: Características generales ACS480	26
Tabla 4: Macros del convertidor de frecuencia	27
Tabla 5: Grupos de parámetros del convertidor de frecuencia	30
Tabla 6: Características técnicas TB511	36
Tabla 7: Características técnicas PM573-ETH	38
Tabla 8: Características técnicas DA501	40
Tabla 9: Características técnicas CD522	42
Tabla 10: Características técnicas CP630	45
Tabla 11: Partes de la maqueta	46
Tabla 12: Configuración IP's	51
Tabla 13: Configuración BCI convertidor	56
Tabla 14: Funciones DI1, DI2 y DI3	69
Tabla 15: Funciones DI4 y DI5	70
Tabla 16: Función DI6	70
Tabla 17: Parámetros nominales del motor	71
Tabla 18: Configuración parámetros ACS480.....	98
Tabla 19: Entradas ACS_COM_MOD_RTU	99
Tabla 20: Entradas ACS_DRIVES_CTRL_STANDARD	101
Tabla 21: Entradas ACS_MOD_READ_N_PRM.....	102
Tabla 22: Entradas ACS_MOD_WRITE_N_PRM	104

1. INTRODUCCIÓN

En un mundo globalizado, la competencia entre empresas se intensifica. En este aspecto, conseguir una ventaja competitiva respecto del resto de empresas es fundamental para poder obtener mejores resultados y, por lo tanto, tener una posición firme en el mercado.

Frente a esta disposición, la automatización industrial representa un papel fundamental en el desarrollo de las empresas. Esta permite mejorar el rendimiento y la fiabilidad de la producción, conocer a tiempo real información del proceso y reducir su tiempo de procesado, optimizar los recursos (energía, materiales,...) y aumentar la seguridad del personal. Todos estos factores permiten a las industrias competir de una forma mucho más efectiva dentro de su propio sector.

En el presente proyecto se implementa un sistema de control y supervisión sobre un motor mediante la interconexión de equipos industriales. Ambas funciones, control y supervisión, se realizan mediante dos fuentes seleccionables que actúan sobre un convertidor de frecuencia. La fuente predeterminada es una maqueta que permite actuar sobre las entradas y salidas físicas del variador de frecuencia. La otra fuente, es un autómatas programable y una interfaz hombre-máquina (HMI) que opera sobre la memoria del convertidor de frecuencia. Las comunicaciones de esta segunda fuente se han realizado mediante un bus de campo integrado que utiliza el protocolo Modbus RS-485. Todo el sistema se ha implementado a partir de equipos y herramientas software de la marca ABB basados en el estándar IEC 61131.

Este documento profundiza en los conocimientos teóricos y prácticos de tecnologías ampliamente extendidas en la industria, con la intención de obtener conocimientos generales en la materia y, por lo tanto, aumentar las capacidades en los estándares más usados.

2. OBJETO Y ALCANCE DEL PROYECTO

El propósito del proyecto es la interconexión de equipos industriales mediante Modbus RTU para la automatización de procesos, como base de desarrollo de proyectos de automatización más complejos. Mediante el enlace de equipos industriales se debe crear una plataforma de supervisión y control de un motor a través de un convertidor de frecuencia.

El control y la supervisión del sistema se tiene que poder ejercer desde dos fuentes seleccionables, o bien desde una maqueta que actúa sobre las entradas y salidas físicas del variador, o bien desde un panel de control táctil y un autómatas que operan sobre la memoria del convertidor.

El control debe proporcionar la posibilidad de realizar la marcha y el paro, seleccionar diferentes velocidades y actuar sobre parámetros del convertidor.

El sistema de supervisión debe permitir controlar a tiempo real el estado del sistema y los parámetros considerados trascendentales.

Los objetivos que se esperan alcanzar al finalizar el proyecto son:

- Documentar y analizar los equipos industriales que forman parte del sistema.
- Estudiar y profundizar en las herramientas software empleadas para el desarrollo de la aplicación.
- Profundizar en el estudio de los protocolos de comunicación, especialmente en Modbus RTU.
- Habilitar el cableado de conexión entre la maqueta y el variador.
- Realizar el cableado del autómatas programable.
- Efectuar la configuración de parámetros del convertidor de frecuencia.
- Configurar las comunicaciones entre los equipos implicados en el proyecto.
- Realizar la programación de las funcionalidades y las comunicaciones del sistema
- Programar una interfaz HMI intuitiva y sencilla que permita la lectura de parámetros y su modificación.
- Comprobar el correcto funcionamiento práctico de todo el sistema, pudiendo realizar el control y la supervisión del motor y el convertidor de frecuencia desde ambas fuentes.
- Realizar un documento del proyecto detallando los equipos y tecnologías empleadas y el trabajo realizado. De modo que este proyecto pueda servir como plataforma de desarrollo para proyectos de automatización más complejos.

3. METODOLOGÍA

3.1. Desarrollo

La sistemática de trabajo utilizada para el desarrollo del proyecto se ha basado en la una primera fase de recopilación de información y una posterior de desarrollo de la aplicación.

La primera fase del desarrollo tiene como objetivo la adquisición de las herramientas software necesarias para el desarrollo de la aplicación y, en la recopilación de información y su posterior análisis.

Una vez adquirida una base de conocimientos sobre la que sustenta esta etapa, se ha desarrollado de manera práctica la aplicación de control y supervisión. Para su realización, se han marcado unos objetivos generales, los cuales, se han desglosado en tareas de medio-corto plazo sobre las que ir trabajando. El orden de estos objetivos ha sido marcado por las necesidades del proyecto y fuertemente condicionado por mi situación personal, es decir, en ocasiones se han tenido que priorizar algunos aspectos sobre otros por tal de compaginar las necesidades del proyecto con la situación comentada.

3.2. Diagrama de Gantt

En la siguiente página se muestra el diagrama de Gantt en el que se representa la planificación que se estimó al inicio del proyecto para su realización.

Se puede observar que la planificación inicial se basaba en una primera etapa de adquisición de las herramientas software y recopilación de la información. Posteriormente se debía haber realizado el cableado del variador con la maqueta y el de los módulos del autómat, una primera configuración básica del convertidor para poder comprobar si el cableado realizado era correcto y establecer la comunicación del PC con el autómat. Durante los meses estivales se debería haber empezado a documentar lo realizado hasta el momento así como el funcionamiento teórico de los equipos. También se estimaba haber realizado durante el mes de julio la comunicación del autómat con el panel de control y durante el mes de agosto el diseño del HMI, por tal de probar su funcionamiento en el mes de Septiembre. A continuación se debería haber iniciado la programación del autómat para su comunicación con el convertidor de frecuencia. Posteriormente se debía realizar la configuración completa del convertidor de frecuencia e iniciar el reescalado de las variables por tal de comprobar el funcionamiento del sistema completo. Los últimos meses se planificaron para realizar ajustes en el sistema y acabar de realizar la memoria.

Acciones	Año 2019									Año 2020			
	Abr	May	Jun	Jul	Ago	Set	Oct	Nov	Dic	Ene	Feb	Mar	Abr
Recopilación información general													
Descarga e instalación software													
Cableado variador-maqueta													
Cableado del autómeta													
Configuración básica convertidor													
Comunicación PC-PAC													
Pruebas variador-maqueta													
Comunicación PAC-HMI (RS-485)													
Programación interfaz HMI													
Comunicación PAC-Convertidor													
Redacción del documento													
Configuración convertidor													
Comunicación PAC-HMI (TCP)													
Reescalado variables convertidor													
Pruebas de conjunto													

Figura 1: Planificación inicial

4. DESCRIPCI3N DEL SISTEMA

El sistema tiene como objetivo realizar el control y la supervisi3n de un motor mediante un convertidor de frecuencia controlado por una maqueta que actúa sobre las entradas y salidas tanto anal3gicas como digitales y, un aut3mata y un HMI que actúan sobre la memoria del variador. Las comunicaciones de estos equipos se realizan a trav3s del protocolo Modbus RTU y mediante el est3ndar RS-485.

Los equipos y sus software para su programaci3n y/o configuraci3n son los siguientes:

Equipo		Versi3n	Software	Versi3n
Convertidor de frecuencia	ACS480	Firmware 2.10.0	Drive Composer	2.3.1
		Panel de mando 5.95		
Aut3mata	AC500 PM573-ETH	2.8.0.0	Automation Builder	2.2.4.631
HMI	CP630	2.31	Panel Builder 600	1.90.0

Tabla 1: Equipos y software

El sistema tiene la capacidad de controlar las acciones del motor y supervisar su estado, al mismo tiempo que permite la parametrizaci3n del control sobre el motor desde el HMI, el aut3mata programable o la maqueta.

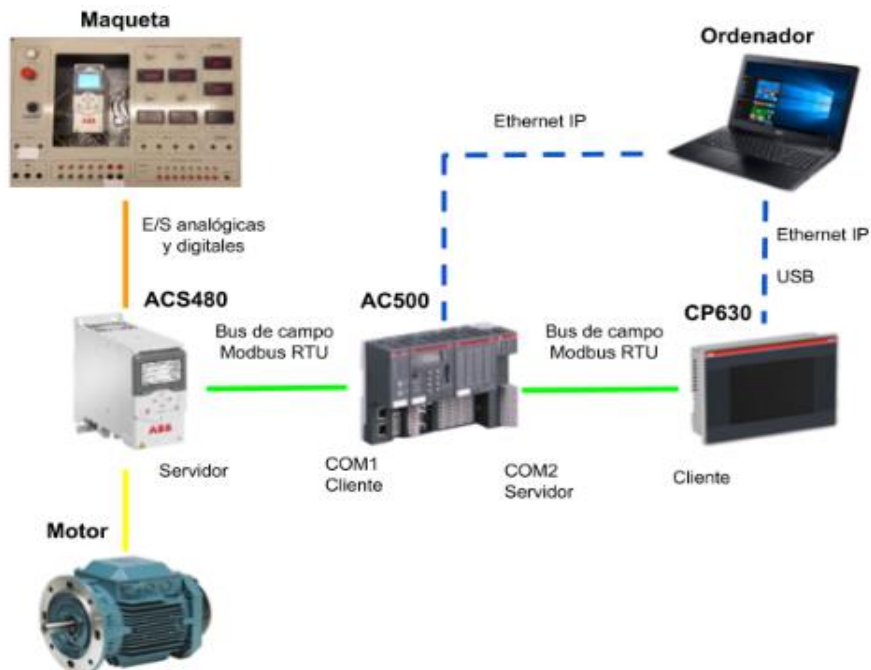


Figura 2: Arquitectura del sistema

5. EL CONVERTIDOR

5.1. Fundamentos teóricos

Los convertidores de frecuencia son dispositivos electrónicos que permiten realizar un control completo de motores eléctricos de inducción mediante el control de la tensión y frecuencia de la alimentación suministrada. Operan bajo el principio de que la velocidad síncrona de un motor de corriente alterna está determinada por la frecuencia y el número de polos en el estator. Controlando la relación tensión y frecuencia, se puede controlar el flujo y de esta forma también el par motor. Los convertidores de frecuencia se sitúan entre la red de alimentación y los bornes del motor. Se pueden clasificar en dos tipos:

- Los convertidores directos o cicloconvertidores están formados por conjuntos de tiristores, los cuales, realizando una serie de conmutaciones sobre la tensión de entrada son capaces de proporcionar una corriente alterna mono o polifásica de amplitud y frecuencia regulables a partir de un generador de corriente alterna. Su principal aplicación se da en el control a baja velocidad de grandes motores de corriente alterna donde se precisa variar la amplitud de la tensión proporcionalmente a la frecuencia.
- Los convertidores indirectos son los variadores de frecuencia más extendidos en la industria. Estos dispositivos no actúan directamente sobre la tensión de entrada, sino que disponen de tres etapas básicas. En primer lugar se encuentra un rectificador que convierte la alimentación en alterna en corriente continua. Posteriormente se encuentra el circuito intermedio o bus de continua, el cual, realiza un filtrado sobre la señal. En la última etapa actúa el inversor, que ondula la señal de continua para convertirla de nuevo en sinusoidal.

El esquema de un convertidor de frecuencia es el siguiente:

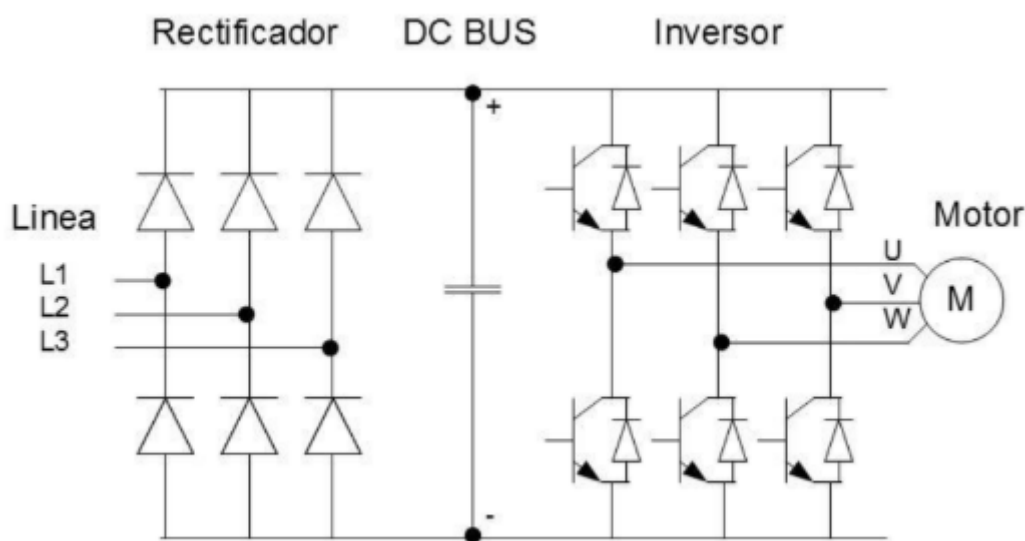


Figura 3: Esquema convertidor de frecuencia indirecto

A continuación se analiza con detalle cada una de las etapas:

Etapas de rectificado: En esta etapa se convierte la corriente alterna en corriente continua. Para ello existen dos tipos de rectificadores, los controlados y los no controlados. Los rectificadores controlados están formados por tiristores, los cuáles, pueden modificar su ángulo de disparo. Por el contrario los rectificadores no controlados están formados por diodos y no pueden modificar su ángulo de disparo. En ambos casos cada diodo o tiristor actúa $\pi/3$ del periodo, convirtiendo así la forma de la onda de entrada en una polaridad constante en la salida.

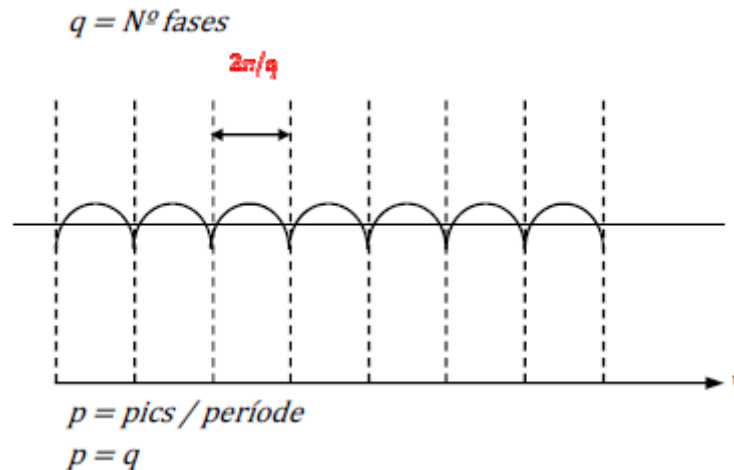


Figura 4: Forma de la tensión después de la etapa de rectificado

Los rectificadores no controlados están ampliamente extendidos en el mercado para aplicaciones de baja potencia, puesto que son innumerables los equipos que se conectan a las redes domésticas trifásicas pero para su funcionamiento utilizan corriente continua.

Etapas de filtrado: En esta etapa se realiza un filtrado de la onda, es decir, se minimiza el rizado de la onda que se ha producido en la etapa anterior. Para ello se emplean condensadores, los cuales, permiten absorber los picos de corriente provocados por la conmutación. La mayoría de variadores tienen una salida accesible al bus de corriente continua y es posible hacer uso de ella como tensión auxiliar. Esto permite acumular energía cuando la carga actúa como generador o en los procesos de frenado.

Etapas de inversión: El objetivo principal de los inversores es producir una salida sinusoidal de corriente alterna cuya amplitud y frecuencia puedan controlarse. Sin embargo, los inversores conmutados permiten que el flujo de potencia sea reversible, y en momentos puntuales donde la carga actúa como freno, por ejemplo, cuando se desacelera un motor de corriente alterna permiten recuperar la energía cinética asociada a la inercia del motor y acumularla en el bus de continua. Los inversores se clasifican según el tipo de entrada, denominándose VSI si es un inversor por fuente de voltaje, o CSI si es inversor por fuente de corriente, siendo este último utilizado solo en aplicaciones con excitadores de motores de alterna de muy alta potencia. Los VSI se subdividen en tres categorías generales:

- Inversores modulados por ancho de pulsos (PWM): En estos inversores, el voltaje de corriente continua de entrada es esencialmente constante y el inversor controla la amplitud y la frecuencia del voltaje. Esto permite mantener el par constante sin desaprovechamiento de la energía eléctrica cuando la carga es un motor de corriente alterna, por ello son los inversores más utilizados en la actualidad. Este tipo de inversores, para producir una forma de onda sinusoidal de voltaje de salida en una frecuencia deseada, comparan una señal de control sinusoidal con la frecuencia deseada con una forma de onda triangular, la cual, establece la frecuencia de conmutación del inversor y por lo general se mantiene constante junto a su amplitud. El voltaje de salida del inversor no es una onda sinusoidal perfecta y contiene componentes de voltaje en frecuencias armónicas.

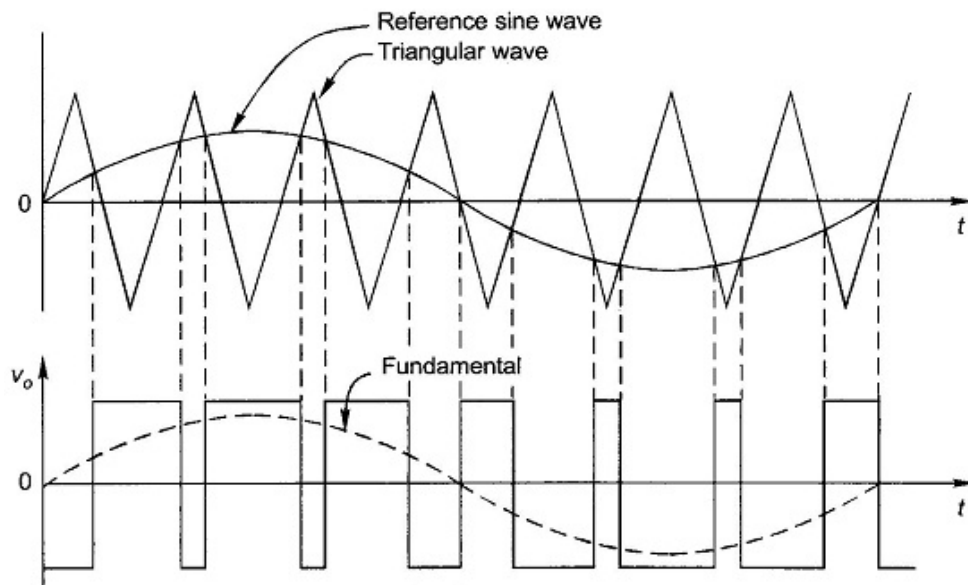


Figura 5: Señal de control triangular comparada con la señal de referencia

- Inversores de onda cuadrada: En este caso, el voltaje de corriente continua de entrada se controla a fin de controlar la amplitud de voltaje de salida, y por tanto el inversor únicamente controla la frecuencia del voltaje de salida. La forma de onda de la salida es parecida a una onda cuadrada y por este motivo estos inversores se denominan de este modo.
- Inversores monofásicos con cancelación de voltaje: En estos inversores con salida monofásica es posible controlar la amplitud y la frecuencia del voltaje de salida del inversor, aunque la entrada al inversor sea un voltaje de corriente continua constante y los interruptores del inversor no se modulen por la anchura de pulsos. Por tanto, estos inversores combinan las características de los dos inversores anteriores. Esta técnica únicamente funciona para inversores monofásicos y no para trifásicos.

5.2. Convertidor ACS480

El convertidor empleado para el desarrollo del proyecto es el modelo ACS480 de la marca ABB. Es un convertidor de frecuencia de propósito general que incluye todos los componentes esenciales para aplicaciones típicas de la industria ligera. Permite el control de motores asíncronos de inducción de corriente alterna, motores de imanes permanentes, motores síncronos de reluctancia. Las aplicaciones más comunes son para el control de compresores, cintas transportadoras, ventiladores y muchas otras aplicaciones básicas de control de velocidad.



Figura 6: Variador de frecuencia ACS480

A continuación se muestran el esquema general del hardware del convertidor y el esquema de las conexiones de control fijas de la unidad base y conexiones de control opcionales basadas en el módulo opcional instalado:

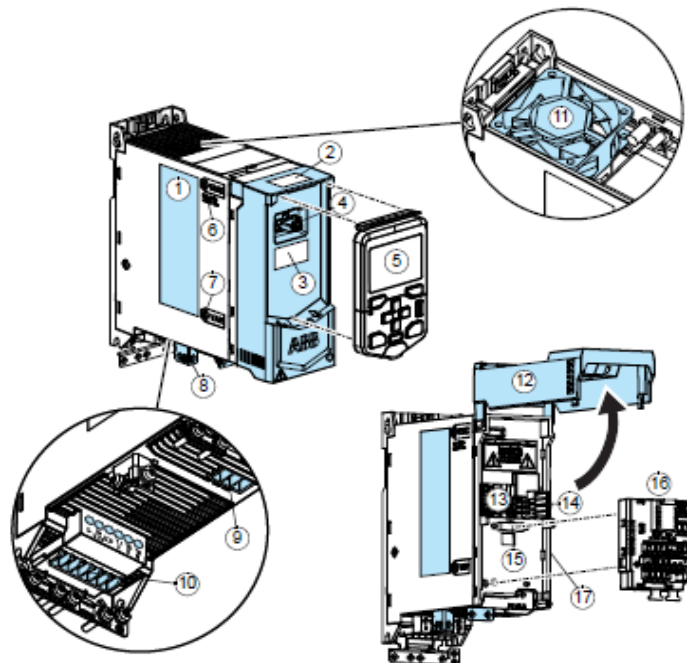
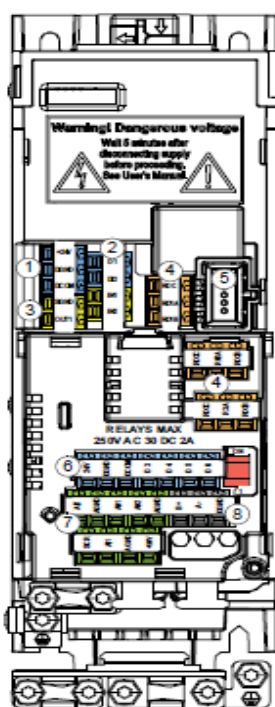


Figura 7: Descripción general del hardware.

Nº	Descripción	Nº	Descripción
1	Etiqueta de designación del tipo	10	Terminales de la resistencia de frenado y del motor
2	Etiqueta de información de modelo	11	Ventilador de refrigeración (no en R0)
3	Etiqueta de información del software	12	Cubierta frontal
4	Conexión del panel de control	13	Terminales de control fijos
5	Panel de control	14	Conexión de configuración en frío
6	Tornillo de conexión a tierra del filtro EMC	15	Ranura de opción de para módulos de comunicación
7	Tornillo de conexión a tierra del variador	16	Módulos de E/S o de bus de campo
8	Conexión PE (motor)	17	Ranura lateral para opcionales de montaje lateral
9	Terminal de potencia de entrada		

Tabla 2: Etiquetas del hardware



Nº	Descripción
1	Salidas de tensión auxiliar
2	Entradas digitales
3	Conexiones Safe Torque Off
4	Conexiones de la salida del relé
5	Conexiones de configuración en frío
6	Entradas digitales
7	Entradas y salidas analógicas
8	EIA-485 Modbus RTU

Figura 8: Esquema de las conexiones de control.

El diagrama simplificado del circuito de potencia del convertidor es el que se muestra a continuación:

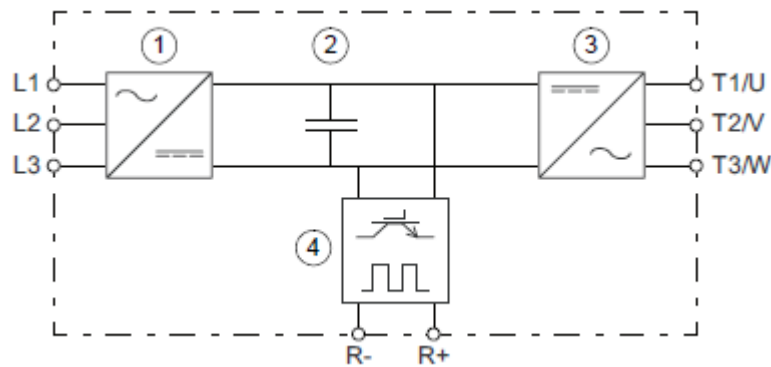


Figura 9: Diagrama simplificado del circuito de potencia del ACS480

El circuito de potencia está formado además de por los elementos de un variador de frecuencia estándar (circuito rectificador, el Bus de CC y el inversor), por un chopper de frenado. Este elemento conduce el exceso de energía del circuito intermedio de CC del convertidor hacia la resistencia de frenado cuando es necesario y si hay una resistencia de frenado externa conectada al convertidor. El chopper funciona cuando la tensión del bus de CC sobre pasa un límite máximo determinado. El incremento de tensión se debe principalmente a la deceleración (el frenado) de un motor.

5.2.1. Características técnicas

Los convertidores de frecuencia ACS480-04 son una gama de dispositivos con características y funcionalidades semejantes. Estos modelos están disponibles para diferentes potencias comprendidas entre 0,75 a 22 kW. En el caso que los requerimientos de potencia fueran mayores, se debería elegir dispositivo de gama superior. Todos los modelos de la gama son estándar, y por lo tanto tienen en común la interfaz de usuario, de proceso de bus de campo y las herramientas de software.

Las características generales de los convertidores de frecuencia ACS480-04, independientemente del modelo son las siguientes:

Conexión a la red eléctrica	
Rangos de tensión y potencia	Trifásico de 380 a 480 V, +10%/-15% 0,75 a 22 kW
Frecuencia	47 a 63 Hz, tasa máxima de variación de 17%/s

Desequilibrio	Máx. $\pm 3\%$ de la tensión nominal de entrada entre fases
Factor de potencia ($\cos \rho$)	0.98 (con carga nominal)
Conexión del motor	
Tipo de motor	Motor de inducción asíncrono o motor síncrono de imanes permanentes
Tensión	Trifásica de 0 a $U_{\text{alimentación}}$
Frecuencia	0 a 599 Hz
Resolución de frecuencia	0,01 Hz
Frecuencia de conmutación	2, 4, 8 o 12 kHz
Conexión de control	
Entradas analógicas	AI1 y AI2
Señal de tensión	0 – 10 V CC (10% sobre el límite) $R_{\text{in}} = 221.6 \text{ k}\Omega$
Señal de intensidad	0 – 20 mA (10% sobre el límite)
Valor de referencia o potenciómetro	10 V CC $\pm 1\%$ intensidad de carga máx. 10 mA
Salidas analógicas	AO1 y AO2
Modo de salida de intensidad	0 – 20 mA en una carga de 500Ω (AO2 solo admite intensidad de salida)
Modo de salida de tensión	0 – 10 V CC en una carga resistiva mínima de $200 \text{ k}\Omega$
Salida de tensión auxiliar /Entrada opcional (+24 V)	
Como salida	+24 VCC $\pm 10\%$ máx. 200 mA
Como entrada	+24 VCC $\pm 10\%$ máx. 100 mA
Entradas digitales	De DI1 a DI6
Tensión	12 – 24 V CC, máx 30 V CC

Tipo	PNP y NPN
Impedancia de entrada	R _{in} = 2kΩ
DI5 (entrada digital o de frecuencia)	12 – 24 V CC, máx 30 V CC PNP y NPN, R _{in} = 2kΩ y frecuencia máx. 16 kHz
Salida de relé	RO1, RO2 y RO3
Tipo	1 de forma C (NA + NC)
Tensión máxima de conmutación	250 V CA / 30 V CC
Intensidad máxima de conmutación	2 A
Entrada de frecuencia	De 10 Hz a 16 kHz
Interfaz STO	Función Safe Torque
EIA-485 Modbus RTU (A+, B-, DGND)	RS-485 Cable de par trenzado apantallado con para trenzado para datos y un cable o par para tierra de la señal, impedancia nominal 100 - 165Ω De 9,6 a 115,2 kbits/s Interruptor
Capa física	
Tipo de cable	
Velocidad de transmisión	
Terminación	
Conexión de la resistencia de frenado	
Protección contra cortocircuitos	La salida de la resistencia de frenado dispone de una protección condicional frente a cortocircuitos según ICE/EN 61800-5-1 y UL 61800-5-1.
Eficiencia	
Eficiencia	Aproximadamente el 98% a potencia nominal
Grados de protección	
Grado de protección (IEC/EN 60529)	IP20 (instalación en armario)

	El convertidor de frecuencia debe instalarse en un armario para cumplir requisitos de protección contra contactos.
Tipo envolvente (UL 61800-5-1)	UL tipo abierto. Sólo para uso en interiores
Categoría de sobretensión (IEC 60664-1)	III
Clases de protección (IEC/EB 60800-5)	I
Condiciones ambientales	
Altitud de lugar de instalación	De 0 a 4000 m sobre el nivel del mar (con derrateo por encima de 1000m).
Temperatura ambiente	De -10 a +60°C
Humedad relativa	De 0 a 95%
Normas aplicables	
EN ISO 13849-1:2015	Seguridad de las máquinas. Partes de los sistemas de mando relativas a la seguridad. Parte 1: Principios generales de diseño.
EN ISO 13849-2:2012	Seguridad de las máquinas. Partes de los sistemas de mando relativas a la seguridad. Parte 2: Validación.
EN 60204-1:2006 + A1:2009 + AC:2010	El montador final de la máquina es responsable de instalar: <ul style="list-style-type: none"> - Un dispositivo de paro de emergencia. - Un dispositivo de desconexión de la fuente de alimentación.
EN 6201:2005 + AC:2010 + A1:2013 + A2:2015	Seguridad de las máquinas. Seguridad funcional de sistemas de mando eléctricos, electrónicos y programables relativos a la seguridad.

EN61800-3:2004 + A1:2012 IEC/EN 61800-5-1:2007	Accionamientos eléctricos de potencia de velocidad variable. Requisitos EMC y métodos de ensayo específicos.
ANSI/UL 61800-5-1:015	Accionamientos eléctricos de potencia de velocidad variable. Requisitos de seguridad eléctricos, térmicos y energéticos.
CSA C22.2 N248 274-13	Accionamientos de velocidad variable.

Tabla 3: Características generales ACS480

5.2.2. Macros y configuración de parámetros

Los variadores de frecuencia ACS480 disponen de perfiles prediseñados llamados llamadas macros. Estas macros modifican un grupo de parámetros en unas variables concretas, lo cual, permite una configuración rápida de sistemas con una finalidad específica. Cuando se modifica una macro, no solamente se seleccionan unos parámetros concretos, sino que el resto vuelven a su valor por defecto. Los valores de los módulos relacionados con la seguridad del sistema no serán reemplazados al modificar la macro.

Las macros disponibles en la gama de dispositivos ACS480 es la siguiente:

Macros	Descripción
ABB estándar	Es la macro por defecto. Ofrece un uso general, una configuración de E/S de 2 hilos con tres frecuencias constantes. Una señal para poner en marcha o parar el motor y otra para seleccionar la dirección. Esta macro usa el control escalar.
ABB estándar (vectorial)	Esta macro usa el control vectorial; por lo demás es similar a la macro ABB estándar que ofrece un uso general.
ABB limitada de 2 hilos	Esta macro usa un número de E/S que solo existen en la unidad base.
3 hilos	Se utiliza cuando la unidad se controla mediante pulsadores momentáneos. Proporciona 3 velocidades constantes.

Alterna	Proporciona una configuración de E/S en la cual una señal pone en marcha el motor en dirección de avance y otra señal pone en marcha el motor en dirección retroceso.
Potenciómetro del motor	Esta macro proporciona un modo de ajustar la velocidad con la ayuda de dos pulsadores, o una interfaz económica para PLC que varían la velocidad del motor únicamente señales digitales.
Manual/Auto	Se puede utilizar cuando se necesite el cambio entre dos dispositivos de control externo. Ambos dispositivos de control externo tienen sus propias señales de referencia y control. Se usa una señal para cambiar entre estas dos.
Manual/PID	Controla el convertidor con regulador PID de proceso incorporado. Además tiene un segundo lugar de control para el modo de control directo de velocidad/frecuencia.
PID	Proporciona ajustes de parámetros para sistemas de control en bucle cerrado, como control de presión, control de caudal, etc.
Panel PID	Esta macro es adecuada para aplicaciones en las que el convertidor siempre está controlado por el PID y el punto de ajuste se define con el panel de control.
PFC	Lógica de control de bombas y ventiladores que permite controlar múltiples bombas y ventiladores mediante las salidas de relé del convertidor.
SPFC	Regula la lógica de control de varias bombas y ventiladores a través de la salida de relé del convertidor.

Tabla 4: Macros del convertidor de frecuencia

La macro seleccionada para llevar a cabo las funcionalidades del proyecto es la macro Estándar ABB. Se ha seleccionado esta macro porque es la que más se ajusta a las necesidades para controlar el motor, dado que en el laboratorio no se tiene una carga mecánica específica a mover. Las asignaciones automáticas (estas asignaciones se pueden personalizar según las necesidades del usuario) de las señales de entrada y salida para esta macro son las siguientes:

Señales de entrada:

- Referencia de frecuencia analógica (AI1)
- Selección de Marcha/Paro (DI1)
- Avance/Retroceso (DI2)
- Selección de frecuencia constante (DI3, DI4)
- Selección de juego de rampas (DI5).

Señales de salida:

- Salida analógica AO1: Frecuencia de salida
- Salida analógica AO2: Intensidad del motor
- Salida de relé 1: Listo para marcha
- Salida de relé 2: En marcha.
- Salida de relé 3: Fallo (-1).

Además de seleccionar las macros, estos variadores, permiten la modificación de parámetros para realizar una configuración ajustada al proceso. Los parámetros están agrupados por familias según su finalidad. Estos se pueden modificar tanto desde el menú del propio variador como por bus de campo (comunicaciones). Los parámetros se definen por direcciones, de modo que cada parámetro tiene su propia dirección. Los dos primeros números definen la familia, y los dos segundos la posición de este parámetro dentro de la familia.

Los grupos de parámetros para los variadores de la gama ACS480 son los siguientes:

Código	Grupo de parámetros
01	Valores actuales
03	Entradas de referencia
04	Avisos y fallos
05	Diagnósticos
06	Palabras de control y estado
07	Información del sistema
10	DI, RO Estándar
11	DIO, FI, FO Estándar
12	AI Estándar
13	AO Estándar

19	Modo de operación
20	Marcha/Paro/Dirección
21	Modo Marcha/Paro
22	Selección de referencia de velocidad
23	Rampas Aceleración/Deceleración velocidad
24	Acondicionamiento referencia de velocidad
25	Control de velocidad
28	Frecuencia cadena de referencia
30	Limites
31	Funciones de fallo
32	Supervisión
34	Funciones temporizadas
35	Protección térmica del motor
36	Analizador de carga
37	Curva de carga de usuario
40	Conjunto PID proceso 1
41	Conjunto PID proceso 2
43	Chopper de frenado
44	Control de freno mecánico
45	Eficiencia energética
46	Ajustes de monitorización/escalado
47	Datos guardados
49	Comunicaciones puerto panel

50	Bus de campo Adaptado (FBA)
51	FBA A Ajustes
52	FBA A Data In
53	FBA A Data out
58	Bus de campo integrado
71	PID1 externo
76	PFC Configuración
77	PFC maintenance and monitoring
95	Configuración hardware
96	Sistema
97	Control del motor
98	Parámetros motor usuario
99	Datos de motor

Tabla 5: Grupos de parámetros del convertidor de frecuencia

5.3. Puesta en marcha del convertidor

Cuando se pone en marcha el convertidor, lo primero que conviene hacer es introducir los parámetros del motor sobre el cual se vaya a realizar el control. Para ello se deben indicar los parámetros nominales del motor en el grupo de parámetros número 99. Una vez introducidos estos valores hay que realizar una marcha de identificación ("Marcha ID"). Es importante que este proceso se realice con el motor libre de carga. De este modo, el convertidor calcula automáticamente las características de motor mediante la magnetización de identificación. Una vez finalizado este proceso, el convertidor crea un modelo a partir de cálculos internos que permite un control eficaz, especialmente en los casos que se realice un control de velocidad o par. Es necesario realizar una marcha de identificación cada vez que se modifica un parámetro nominal del motor.

6. EL AUTÓMATA PROGRAMABLE

6.1. Fundamentos teóricos

Un autómata programable es un sistema electrónico programable diseñado para ser utilizado en un entorno industrial, que utiliza una memoria programable para el almacenamiento interno de instrucciones orientadas al usuario, para implementar unas soluciones específicas tales como funciones lógicas, secuencia, temporización, recuento y funciones aritméticas con el fin de controlar mediante entradas y salidas, digitales y analógicas diversos tipos de máquinas o procesos.

En sus inicios recibían el nombre de PLC's (Programmable Logic Controller) porque en sus inicios procesaban exclusivamente señales digitales y, en consecuencia, únicamente realizaban operaciones lógicas básicas. Además, eran dispositivos compactos que difícilmente era posible modificar su configuración.

En la actualidad, los autómatas programables (PAC – Programmable Automation Controller) tienen la capacidad de trabajar con señales analógicas, PID's, comunicaciones, o incluso, funciones matemáticas avanzadas. Asimismo, no solo han sufrido cambios internos en sus capacidades, sino que los modelos que se encuentran en el mercado actual, suelen ser modulares, lo cual, permite adaptar un dispositivo a las necesidades de un proyecto.

En función de la gama de un autómata se pueden encontrar diferentes prestaciones. Sin embargo, independientemente del fabricante, de la gama y el modelo, los bloques básicos que componen un autómata son los siguientes:



Figura 10: Estructura básica de un autómata programable. Recuperado de <https://www.palentino.es/blog/>

- Fuente de alimentación: Se encarga de suministrar y supervisar la tensión de los diferentes módulos que componen el sistema. Por lo general, el módulo se instala en el primer alojamiento del rack y suele integrar los siguientes elementos:
 - Bloque visualizador: Ofrece información del funcionamiento del módulo y suele incluir indicadores de estado del autómatas y de las comunicaciones con los periféricos.
 - Botón pulsador reset: Permite reiniciar el equipo cuando este está en funcionamiento y se acciona.
 - Bornero de tornillos: Se utiliza para la conexión de la tensión de red y una salida de 24 voltios de continua para alimentar los sensores sin necesidad de alimentación externa.
 - Contacto relé de alarma: Cuando el autómatas está en funcionamiento normal (modo RUN), el relé se mantiene accionado y cerrado, pero cuando se detecta un fallo y/o alarma este se abre.
- Procesador: Es la unidad central de proceso (CPU) y sus funciones son ejecutar los programas, testear la integridad del equipo, guardar y almacenar datos y comunicarse con los periféricos. El elemento principal de este módulo es el procesador. Las características de este módulo son las que determinan las diferentes capacidades que tendrá el autómatas, tales como el conjunto de instrucciones de programación, la capacidad de la memoria, tipos de variables con las que poder trabajar, número máximos de entradas y salidas que se pueden instalar en el sistema y la velocidad del sistema. Las funciones que tiene encomendado este módulo son básicamente las siguientes:
 - Captar el estado de las entradas y almacenarlo en la zona de memoria imagen de las entradas, ya que el programa de usuario no debe acceder directamente a dichas entradas.
 - Ejecutar el programa usuario.
 - Vigilar el tiempo de ejecución de programa para que no exceda de un determinado tiempo máximo. Función denominada watchdog.
 - Renovar el estado de las salidas en función de la imagen de las mismas, obtenida a final del ciclo de ejecución del programa usuario.
 - Revisar el sistema para determinar la existencia de errores de hardware o software.

Con el fin de que el procesador pueda llevar a cabo su trabajo, el autómatas dispone de diferentes tipos de memoria en función de la aplicación a la que está destinada:

- Memoria RAM (memoria de acceso aleatorio): Generalmente es de tipo volátil y muy rápida, se utiliza para almacenar los programas y datos que permiten el

control de procesos. Suele llevar asociada una batería que permite salvaguardar los datos. Actualmente puede llevar flash (sin batería). Esta clase de memoria, también denominada memoria de usuario, está dividida en función de la tarea que tiene asignada en los siguientes tipos:

- Memoria de programa: En ella se almacena el programa que el usuario ha creado y transferido.
 - Memoria del registro de imagen de entrada: Es la zona de memoria donde se almacena el estado lógico de las entradas.
 - Memoria del registro de imagen de salida: Es la zona de memoria donde se almacena el estado lógico de las salidas.
 - Memoria de variables: Es el lugar de la memoria donde se almacenan las diferentes variables internas disponibles en el PLC:
 - Memoria de almacenamiento temporal: Memoria donde se almacenan los datos intermedios y que no son relevantes para el usuario pero son necesarios para realizar los cálculos y las operaciones lógicas del sistema.
- Memoria ROM (memoria de solo lectura): Se trata de la memoria donde se encuentra el sistema operativo del PLC, también denominado firmware. Conjuntamente con el procesador, el firmware determina las funcionalidades del sistema (lenguajes de programación, conjunto de instrucciones, control de fallos, tipos de variables del sistema, etc.). Este tipo de memoria se puede sustituir por una EPROM (erasable programmable read only memory), o cualquier otro tipo de memoria en la que se puedan borrar los datos y reescribirlos, con el fin de actualizar el firmware del sistema.

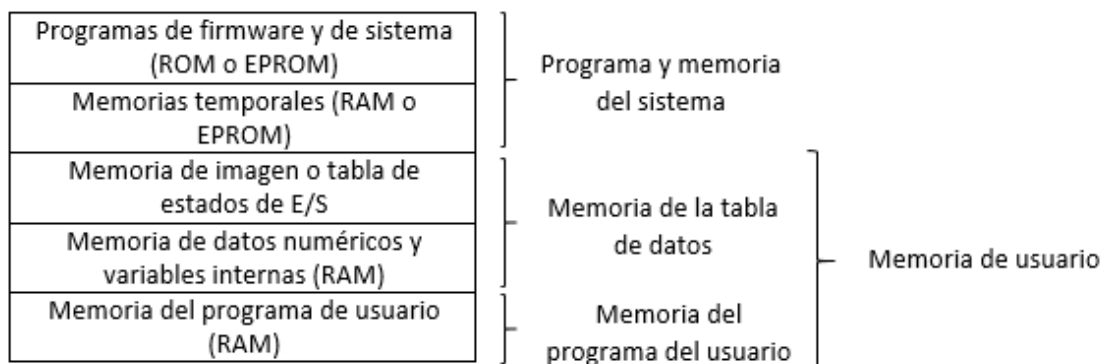


Figura 11: Mapa de memoria de un PLC. Recuperado de *Introducción a los autómatas programables ESEIAAT (UPC)*.

- Módulos de entradas y salidas (E/S): Coordinan la entrada y salida de las señales, con las internas del PLC. Tienen la función de recoger el estado de los distintos captadores, pulsadores y elementos de diálogo (módulo de entradas), y dar señal a los preactuadores y elementos de diálogo hombre-máquina (módulos de salida). La variedad de estos módulos se adapta a la funcionalidad requerida para la instalación, aunque los más habituales son los módulos para el control de dispositivos binarios de automatización. En el mercado se pueden encontrar una gran tipología de módulos, módulos de contaje rápido, de E/S analógicos, de posicionamiento de ejes, de control fuzzy, de comunicación Ethernet,...

Las funciones de los módulos de entradas digitales que integran la estructura del PAC son captar la información binaria procedente de los captadores y elementos de diálogo del proceso, aislar eléctricamente la parte del proceso a controlar otros módulos que integren la estructura del autómatas y adaptar los diferentes niveles de señal del proceso a la tipología de señal que es capaz de trabajar el procesador. Los módulos de entrada digital detectan la presencia o ausencia de tensión en sus bornes de entrada y las convierten en una señal lógica 1 o 0 que se almacena en el registro de entradas del PAC, y mediante este valor lógico, asociado a una de las variables del sistema, se obtiene la información del mismo.

Los módulos de salida permiten enviar información a los procesos en función de los resultados del programa del usuario. La función de los módulos de salidas digitales que integran la estructura del PAC se podría resumir en transferir los resultados lógicos del programa a los diferentes preactuadores y actuadores que conforman el proceso, aislar eléctricamente la parte del proceso a controlar de otros módulos que integren la estructura del autómatas y adaptar las diferentes señales de proceso o tipología de señal que es capaz de trabajar el procesador. Para estos módulos se implementan diferentes tecnologías en función de la aplicación a la que se van a dedicar, la cuales se podrían resumir en salidas por transistor en conmutación, salidas tipo relé y salidas tipo Triac.

- Bus del sistema: Son las líneas eléctricas por las que circulan los datos que permiten la transferencia de los mismos en todas y cada una de las partes del sistema.
- Puerto de conexiones y comunicaciones: Está ligado al procesador del sistema, que permite la conexión de un terminal de programación específico o, lo que es más habitual, conectarlos a un puerto serie o USB de un ordenador configurado con un programa de aplicación que permite programar, transferir y supervisar el programa de aplicación del proceso a controlar, generalmente ubicado en el módulo procesador. También se pueden conectar periféricos auxiliares tales como pantallas táctiles, registradores gráficos o elementos similares, así como incluirlos en una red de comunicaciones de elementos industriales para formar lo que se denomina un bus de campo.

Según la configuración física y características de cada uno de estos módulos, los autómatas se clasifican en diferentes gamas según su funcionalidad (procesador, memoria, programación y número de puntos de E/S:

- Microautómatas: Es la gama más baja de autómatas. Los módulos que conforman el autómata viene prefijados y no son susceptibles de ser manipulados, modificados ni ampliados por el usuario. Suelen tener un número limitado de entradas y salidas, poca memoria RAM y tienen integrado el conector de programación en el propio encapsulado, o bien son programados directamente desde la consola de programación que incorporan en el panel frontal encapsulado.
- Autómatas compactos: Los autómatas compactos, son los denominados de gama baja. Generalmente se componen de una estructura compacta, en un sistema que incorpora fuente de alimentación, procesador y un conjunto limitado de E/S. Inicialmente esta era un estructura cerrada, no modificable ni ampliable, pero actualmente la tendencia es que estos pequeños autómatas incorporen las conexiones de expansión necesarias para poder conectar módulos adicionales, limitado a un número máximo establecido por el fabricante.
- Autómatas modulares: Son los autómatas programables de la gama alta. El autómata se constituye totalmente con elementos modulares (fuente de alimentación, procesador, módulos de E/S, módulos de comunicación,...), de tal manera que permiten una gran flexibilidad en su construcción y en la modificación de su estructura en función de las necesidades del usuario.

6.2. Autómata programable AC500

Los autómatas programables de la marca ABB se disponen en la gama AC500. Esta serie de autómatas programables modulares de gama alta se caracteriza por la flexibilidad a la hora de intercambiar módulos de los diferentes modelos gracias a su montaje en base de terminales, los cuáles, están diseñados para permitir una rápida conexión y desconexión sin la necesidad de instalaciones, cableado o herramientas. El tipo de base depende del número de módulos y las comunicaciones que requiera el proyecto. También permiten integrar canales digitales configurables y una gran variedad de comunicaciones.



Figura 12: Autómata programable de la gama AC500

6.2.1. Terminal base TB511

La base de terminales para conectar la CPU y los módulos extensibles del PAC empleado para la realización del presente proyecto es el modelo TBB511_ETH. Dispone de la posibilidad de integrar un módulo de procesador y un módulo de comunicaciones.



Figura 13: Terminal base TB511

La información técnica de esta base de terminales es la siguiente:

Parámetro	Valor
Conector de la fuente de voltaje 24 VDC al terminal base del módulo procesador.	Bloque extraíble de terminales de 5 pines.
Consumo de corriente máximo a 24 VDC	0,35 A
Ranuras	TB511: 1 módulo procesador, 2 módulo de comunicaciones
Interfaces del módulo procesador	I/O bus, COM1, COM2, FBP
Interfaces del módulo de comunicaciones	TB511-ETH / PM573-ETH: Ethernet
Peso neto (del terminal base sin el módulo procesador)	174 g
Posición de montaje	Horizontal o vertical

Tabla 6: Características técnicas TB511

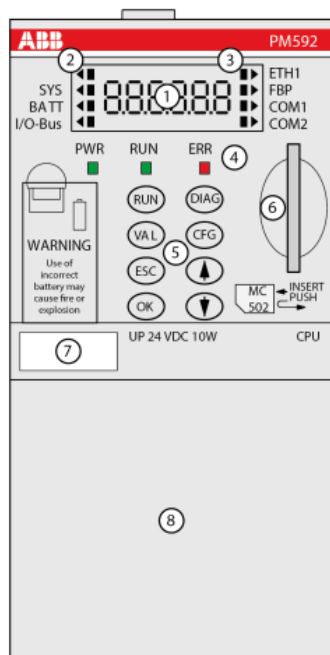
6.2.2. PAC PM573 - ETH

El autómata programable empleado para la realización del proyecto es el modelo PM573-ETH de la marca ABB. Es un autómata modular de gama alta, con tiempos de ciclo rápidos y una tarjeta de memoria que almacena información de usuario y de programa además de las actualizaciones de firmware. El número máximo de módulos de extensión es de 10 unidades y permiten un máximo de 320 E/S digitales y 160 E/S analógicas. En cuanto a las comunicaciones, dispone dos interfaces serie configurables para Modbus RS-232 y Modbus RS-485 (maestro/esclavo) y comunicaciones serie ASCII, además de un módulo de comunicaciones Ethernet. También permite su programación en los lenguajes de programación más utilizados como texto estructurado (ST), lista de instrucciones (LI), bloques de funciones (FB), diagrama de contactos (LD), tabla de funciones secuenciales (SFC), y gráfico de funciones continuas (CFC).



Figura 14: PM573-ETH

El esquema de las funcionalidades e indicaciones físicas que permite el autómata son las siguientes:



Funcionalidades e indicaciones físicas del PM573-ETH	
1	Indicaciones de estado
2	Indicaciones cuadradas para "Ítem"
3	Indicaciones cuadradas para "Estado"
4	LEDs de estado
5	Teclas de función
6	Ranura tarjeta de memoria SD
7	Etiqueta
8	Conexiones base de terminales

Figura 15: Funcionalidades e indicaciones físicas PM573-ETH

Las características técnicas del autómatas PM573-ETH se detallan a continuación:

Característica	PM573-ETH
Tensión Nominal	24V DC
Potencia	10 W
Memoria Flash EPROM and RAM	512 kB
Tarjeta de memoria	Mínimo 512 kB
Tiempo de ciclo por instrucción	
Binaria	0,06 μ s
Word	0,09 μ s
Coma flotante	0,07 μ s
Número máximo E/S	
Entradas digitales	320 unidades
Salidas digitales	24 unidades
Entradas analógicas	160 unidades
Salidas analógicas	160 unidades
Ejecución del programa	
Cíclico, tiempo controlado, tiempo real de reloj, multitarea	
Conexiones	
COM 1	RS232/RS485 configurable
COM 2	RS232/RS485 configurable
FieldBus Plug	PROFIBUS, CANopen, DeviceNet
Ethernet	RJ45

Tabla 7: Características técnicas PM573-ETH

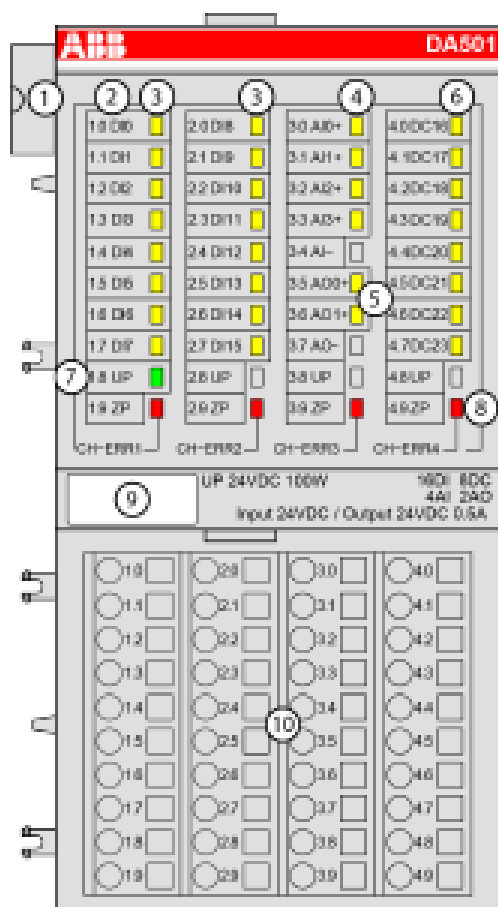
6.2.3. Módulo DA501

El ABB DA501 es un módulo de expansión mixto de entradas y salidas digitales y analógicas. Dispone de 8 canales digitales, los cuáles se pueden configurar tanto como entradas como salidas en función de las necesidades del usuario. Las entradas analógicas también pueden ser configuradas como entradas digitales. También tiene limitación de salida de corriente y reactivación automática después de corto circuito o sobrecarga. El bus permite establecer conexiones con transmisiones de datos fiables de hasta 1000m cuando este es apantallado.



Figura 16: Módulo de expansión DA501

El esquema de funcionalidades e indicaciones físicas para este módulo de expansión son las siguientes:



Funcionalidades e indicaciones físicas del DA501	
1	Bus de entradas y salidas
2	Asignaciones de número de terminal
3	16 LED amarillos para indicar los estados de las entradas digitales
4	4 LED amarillos para indicar los estados de las entradas analógicas
5	2 LED amarillos para indicar los estados de las salidas analógicas
6	8 LED amarillos para indicar el estado de las E/S digitales configurables
7	1 LED verde para indicar el estado de la tensión de proceso UP
8	4 LED rojos para indicar errores
9	Etiqueta
10	Unidad de terminales

Figura 17: Funcionalidades e indicaciones físicas DA501

Las características técnicas del módulo DA501 para la versión estándar de la gama de autómatas AC500 son las que se detalla a continuación:

Entradas Digitales	
Número de canales	16 canales repartidos en 2 módulos
Terminales de los canales	De 1.0 a 1.7 y de 2.0 a 2.7
Entradas/Salidas Digitales Configurables	
Número de canales	8 canales
Terminales de los canales	De 4.0 a 4.7
Entradas Analógicas	
Número de canales	4 canales
Terminales de los canales	De 3.0 a 3.3
Salidas Analógicas	
Número de canales	2 canales
Terminales de los canales	De 3.5 a 3.6

Tabla 8: Características técnicas DA501

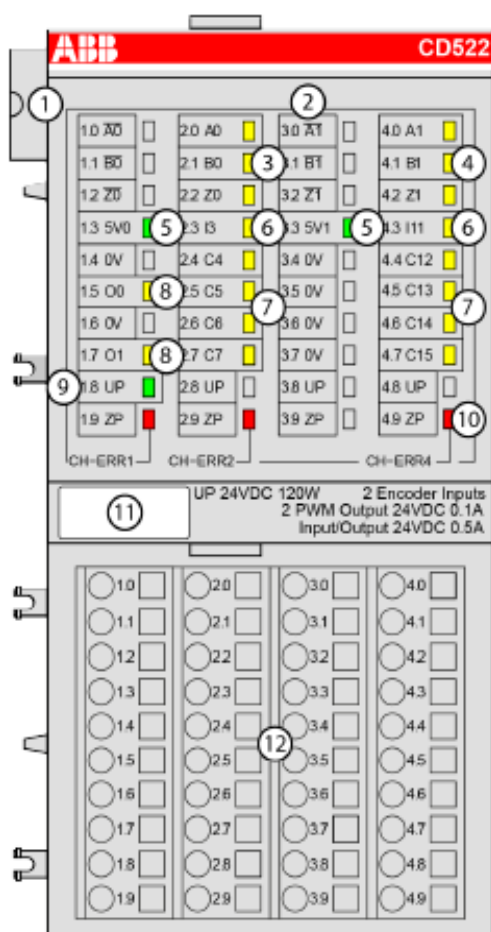
6.2.4. Módulo CD522

El CD522 es un módulo de expansión de encoder, contador y señales PWM, lo que permite realizar el control en tiempo real de servoaccionamientos comandados desde protocolos de comunicaciones. Está formado por 2 entradas encoder, 2 salidas PWM, 2 entradas digitales y 8 canales digitales configurables como entradas y salidas. Las entradas encoder están alimentadas a 24 V de continua y cada una de ellas trabaja independientemente de la otra. Su configuración se realiza mediante el software Automation Builder. El módulo se reactiva automáticamente después de corto circuito o sobrecarga y la longitud máxima para realizar transmisiones sin pérdidas de datos varía en función de las entradas y salidas que se empleen.



Figura 18: Módulo de expansión CD522

El esquema de funcionalidades e indicaciones físicas para este módulo de expansión son las siguientes:



Funcionalidades e indicaciones físicas del CD522	
1	Bus de entradas y salidas.
2	Asignaciones de número de terminal y nombre de señal.
3	3 LED amarillos para indicar los estados de las señales de la entrada del codificador 0.
4	3 LED amarillos para indicar los estados de las señales de la entrada del codificador 1.
5	2 LED verdes para indicar los estados de alimentación de 5V.
6	2 LED amarillos para indicar el estado de señal de las entradas digitales I3 e I11.
7	8 LED amarillos para indicar los estados de las señales entradas/salidas.
8	2 LED amarillos para indicar los estados de las señales de las salidas de los pulsos PWM.
9	1 LED verde para indicar la tensión de proceso UP
10	3 LED rojos para indicar errores
11	Etiqueta
12	Unidad de terminales

Figura 19: Funcionalidades e indicaciones físicas CD522

Las características técnicas del módulo CD522 para la versión estándar de la gama de autómatas AC500 son las que se detallan a continuación:

Entradas Digitales	
Amplitud de la señal de entrada	24 V
Amplitud para señal 0	-3 a 5V
Amplitud señales indefinidas	5 a 15V
Amplitud para señal 1	15 a 30V
Longitud de los datos de entrada	24 bytes
Salidas Digitales	
Amplitud de la señal de salida	500mA por cada canal que este UP
Valor máximo	8 A
Longitud de los datos de salida	32 bytes
Retardo en el cambio de estado de las señales	10µs
Frecuencia de conmutación máxima	Cargas resistivas: 0Hz Cargas inductivas: 0.5Hz LEDs: 11Hz
Encoder	
Voltaje de salida para señal 0	≤ 1.5 V a 10 mA
Retardo conmutación estado	0.3 µs
Frecuencia de conmutación	< 1 MHz (depende del firmware)
PWM	
Voltaje de salida para señal 1	UP (-0.1 V)
Voltaje de salida para señal 0	ZP (+0.3 V)
Retardo en la salida	1 µs
Corriente de salida por canal activo	100 mA
Valor máximo de corriente de salida	8 A
Frecuencia de conmutación	PWM hasta 100KHz

Tabla 9: Características técnicas CD522

7. HMI

7.1. Fundamentos teóricos

Un sistema HMI (humano-machine interface) es una plataforma de monitoreo industrial diseñada para actuar como puente de comunicación entre máquinas y personas, es decir, permite la interacción entre un proceso y cualquier sistema que precise de la acción de una persona. Las pantallas HMI se utilizan para optimizar procesos industriales digitalizando y centralizando los datos a tiempo real. De este modo los operadores pueden visualizar los datos que se consideren más relevantes, gestionar alarmas, y conectarse con sistemas SCADA a través de una consola.

Los sistemas HMI se comunican con los controladores lógicos programables y los sensores de las entradas/salidas para obtener y mostrar información mediante recursos gráficos, de modo que proporciona al operador funciones activas a realizar sobre los resultados de esas acciones e información sobre el rendimiento del sistema.

Generalmente los sistemas HMI se utilizan conjuntamente con sistemas SCADA en la industria de la automatización (el HMI se utiliza para monitorear o visualizar lo ejecutado por SCADA). SCADA son las siglas de Sistemas de Supervisión, Control y Adquisición de Datos y es un sistema de automatización capaz de monitorear, controlar y enlazar dispositivos. A través de servidores puede almacenar datos, estadísticas historiales de control de procesos, elementos de seguridad y hasta procesos en la nube.

En la industria donde la conectividad, la adquisición de datos, su tratamiento, visualización y almacenamiento es vital, un sistema como el HMI SCADA satisface las exigencias de la nueva era industrial. Algunas de las ventajas de la implementación de estos sistemas se podrían resumir en reducción de costes operacionales, análisis del sistema para su mejor continua, aumento de la calidad por disminución de errores evitables, simplificación del control de procesos, monitorización de datos a distancia, y simplificación de la interacción entre sistemas.

7.2. Panel CP630

El panel de control CP630 es un terminal de operador táctil de la gama de HMI de ABB. Este dispositivo permite el monitoreo y control de procesos mediante su programación con el software Panel Builder 600.



Figura 20: CP630

Este terminal, no dispone de botones para su mando, únicamente de puertos de comunicación para cargar los programas y establecer las comunicaciones con el resto de equipos. En cuanto a sus dimensiones, tiene una pantalla táctil de 5,7" y sus dimensiones totales son de 187x147x47mm con un peso de 1,15 Kg.

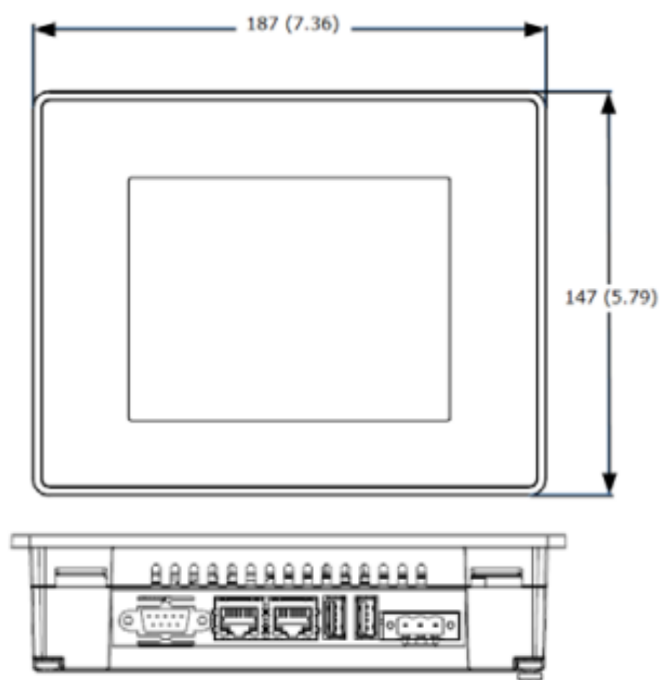


Figura 21: Dimensiones y puertos del panel CP630

A continuación se muestran las características técnicas del módulo CP630:

Información de la pantalla	
Tamaño de la pantalla	5,7"
Resolución	320 x 240 pixeles
Material del soporte	Aluminio y zamac
Material de la pantalla	Cristal cubierto de película de plástico
Recursos	
Procesador	ARM Cortex-A8, 600 MHz
Sistema operativo	Microsoft Windows 6.0 core
Memoria de aplicaciones	Proyectos de hasta 30 MB
RAM	256 MB
Flash	128 MB
Ranura para tarjeta	Tarjeta SD
Interfaces de comunicación	
Puertos Ethernet 0 y 1	10/100 Mbit
COM1	RS-232/485/422 configurable por software
Puerto USB 1	Interfaz servidor V2.0
Puerto USB 2	Interfaz servidor V2.0 y V1.1.
Fuentes de alimentación	
Alimentación V DC	24 V DC (18 – 32 V DC)
Consumo a 24 V DC	0.7 A
Conectores	Terminal de 3 pin con conexión
Grado de protección	IP20

Tabla 10: Características técnicas CP630

8. MAQUETA

Para el control y monitorización del convertidor de frecuencia además del HMI, se ha adaptado una maqueta para el variador ACS480. Para ello se ha realizado un mecanizado en la apertura de la chapa frontal, se ha instalado el nuevo variador de frecuencia, se ha cableado, y se han adaptado el número de entradas y salidas tanto analógicas como digitales de la maqueta a las del variador. Se han anulado las DI 7 y 8, las AI 3 y 4 y la AO 3.

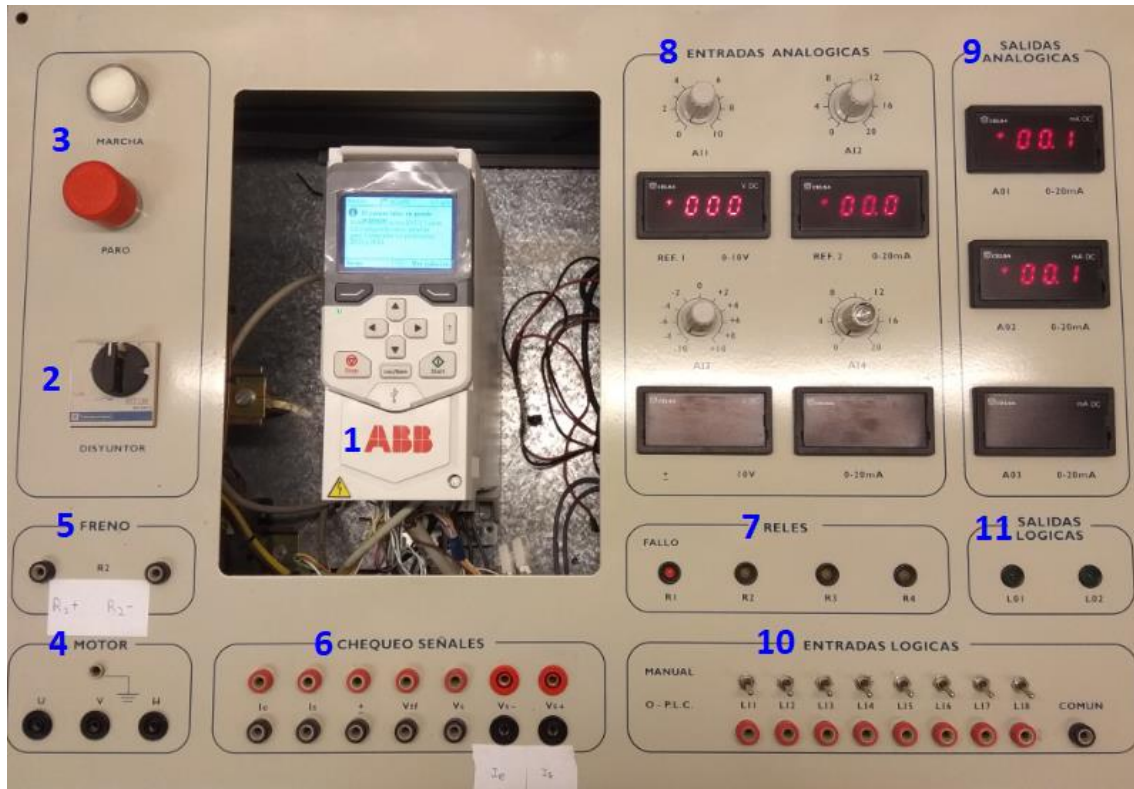


Figura 22: Maqueta

En dicha maqueta pueden distinguirse diferentes partes que son las que se enumeran a continuación:

Nº	Descripción	Nº	Descripción
1	ACS480	7	Relés
2	Disyuntor	8	Entradas analógicas
3	Pulsadores de marcha y paro	9	Salidas analógicas
4	Bornes del motor	10	Entradas digitales
5	Bornes de la resistencia de frenado	11	Salidas digitales
6	Chequeo de señales		

Tabla 11: Partes de la maqueta

9. PROTOCOLOS DE COMUNICACIÓN INDUSTRIAL

Un protocolo de comunicaciones es un sistema de reglas y procedimientos que permiten que dos o más nodos de una red se comuniquen entre ellos para transferir e intercambiar datos. Se trata de las reglas que definen la sintaxis, semántica y sincronización de la comunicación, así como los métodos de detección y recuperación de errores.

Los sistemas de comunicación utilizan formatos definidos (protocolo) para intercambiar mensajes. Cada mensaje tiene un significado exacto destinado a obtener una respuesta de un rango de posibles respuestas predeterminadas. Las entidades involucradas en la transmisión de datos deben utilizar el mismo protocolo, el cual, debe ser desarrollado dentro de un estándar técnico.

9.1. Modbus

Modbus es un protocolo de transmisión para sistemas de control y supervisión de procesos con control centralizado, que puede comunicarse con una o varias estaciones remotas (RTU), con la finalidad de obtener datos para la supervisión y control de un proceso. Las interfaces de capa física pueden estar configuradas en RS-232, RS-422 y RS-485.

Modbus es un protocolo de tipo petición/respuesta, por lo que cuando se produce un intercambio de datos se identifica al dispositivo que realiza la petición como cliente o maestro, y al que devuelve la respuesta como servidor o esclavo de la comunicación. En una red Modbus el equipo maestro es el que tiene acceso a varios equipos catalogados como esclavos. Cada uno de estos dispositivos esclavo se identifica con una dirección única en la red.

En Modbus los datos pueden intercambiarse en 3 modos de transmisión: RTU, ASCII o TCP. Los dos primeros están diseñados para ser utilizados directamente sobre un medio físico serie asíncrono como por ejemplo RS-485. En cambio, el modo TCP está desarrollado para funcionar sobre redes que utilizan la arquitectura TCP/IP, lo que permite usar Modbus sobre redes Ethernet o wifi.

9.1.1. Modbus RTU

Este protocolo se desarrolló para la comunicación de PLC, pero también se utiliza para el control básico de parámetros en las unidades, es uno de los bus de campo más utilizados.

Modbus RTU dispone de una topología física punto a punto y topología lógica maestro/esclavo sobre RS-485. Se caracteriza porque dispone de un buen aprovechamiento del canal de comunicación, mejorando la velocidad de la transmisión de datos aunque esta depende del enlace serie. Pero por el contrario requiere una gestión de tiempos entre bytes para saber cuándo empiezan y terminan las tramas. Permiten la conexión de un máximo de 512 dispositivos, una longitud máxima en estándares RS-485 de 1200 metros y el bus de comunicaciones empelado debe ser una cable de par trenzado apantallado o no.

9.1.2. Modbus TCP

La especificación Modbus TCP permite encapsular el paquete Modbus estándar dentro de la estructura de mensajería TCP/IP. Este es el protocolo Ethernet industrial más popular debido a la sencillez de la aplicación utilizando Ethernet. Dispone de velocidades de transmisión altas, topología física en estrella por lo general, pero se pueden emplear otras topologías como bus, anillo o árbol y utiliza la arquitectura maestro/esclavo. El número de nodos y distancia es teóricamente ilimitado, pero en la realidad depende de la velocidad de transmisión.

10. HERRAMIENTAS SOFTWARE

Este capítulo introduce al lector en las capacidades y funcionalidades de las herramientas software empleadas en el desarrollo del proyecto. Para la descarga, instalación e introducción al entorno se dispone de un capítulo en los anexos.

ABB ofrece herramientas de software integradas para sus dispositivos que facilitan y mejoran su uso además de aumentar su productividad y disminuir el coste de la instalación. Generalmente estos software permiten la creación de proyectos, a través de los cuáles se pueden establecer configuraciones predeterminadas en los dispositivos y plataformas de programación y simulación para recrear su funcionamiento. Todo esto se puede realizar sin la necesidad de disponer de los equipos a nivel físico.

10.1. Drive composer

Drive composer es una herramienta de puesta en marcha y mantenimiento para los convertidores de arquitectura común de ABB. Se usa para visualizar y definir los parámetros del convertidor, así como monitorizar y ajustar el rendimiento de procesos.

La versión básica de Drive composer que ha sido la empleada en la realización del proyecto, ofrece las funcionalidades básicas para el ajuste de parámetros, la monitorización básica, controlar localmente el convertidor desde el PC y gestionar el registrador de eventos. Esta versión está disponible de forma gratuita y se puede descargar desde la página web de ABB.

La versión Premium llamada Drive composer pro presenta las funciones de la versión básica y además incluye diagramas de control, monitorización rápida, funcionamiento con varios convertidores en la red de herramientas de PC y edición de scripts de macros para parámetros.

10.2. Automation Builder

Automation Builder es una plataforma de programación integrada utilizada ampliamente en la industria de la automatización que permite desarrollar proyectos para autómatas, paneles de control, convertidores, y robots desde un mismo entorno. Por lo tanto, permite definir en un único archivo todos los dispositivos que conforman el sistema.

Este software dispone de las herramientas necesarias para la configuración, la programación, la depuración, el diagnóstico y la simulación. La programación de autómatas mediante Automation Builder está basada en el software de programación Codesys, por lo tanto, toda programación se realiza mediante esta plataforma.

10.2.1. Codesys

Codesys es un acrónimo de *Controller Development Systems* que significa Sistema de Desarrollo de Controladores. Es un software de código abierto gratuito que se puede descargar de la página web del fabricante. Es un programa versátil que da la posibilidad de trabajar con dispositivos de distintos fabricantes y combinarlos entre ellos. También permite la integración de bibliotecas o ampliaciones de software según la funcionalidad del proyecto, así como la adaptación a sistemas ya existentes o estándares OPC y protocolos de comunicación diversos.

Es un entorno de programación basado en el estándar industrial internacional IEC-61131-3 y está destinado principalmente a la programación de autómatas y reguladores industriales, pero también permite la programación de microprocesadores.

Este entorno permite la programación combinada en los 5 lenguajes de programación estándar, la depuración del código para la detección de errores y su simulación offline para verificar el correcto funcionamiento. Para la monitorización y el control de los programas desarrollados dispone de un entorno de visualizaciones interno semejante a un SCADA.

10.3. Panel Builder

El software Panel Builder 600 permite la programación y el diseño de paneles de control de la marca ABB además de la comunicación con otros dispositivos mediante un gran número de protocolos de comunicaciones.

Este software dispone de una gran variedad de recursos. Permite la adquisición de datos, su representación en múltiples formatos, gestión de usuarios con contraseñas y permisos, sistemas multilenguaje, históricos de alarmas, múltiples capas y menús rotativos.

Los objetos visuales tienen un formato bidimensional y se les pueden asociar animaciones para mejorar la interactividad con el usuario. Estos objetos son de código abierto y por lo tanto permiten un alto nivel de personalización y flexibilidad.

En la versión premium de Panel Builder 600 es posible integrar los proyectos con Automation Builder, lo cual, permite programar ambos proyectos desde un mismo entorno. En el caso que únicamente se disponga de la versión básica, los proyectos deberán programarse por separado.

11. COMUNICACIÓN

Para poder establecer comunicaciones entre los diferentes dispositivos que forman parte del presente proyecto, es necesario establecer la correcta configuración del bus de campo seleccionado en cada uno de ellos. Los parámetros configurados en los diferentes dispositivos varían en función del protocolo empleado y de los equipos que se desean comunicar.

11.1. Comunicación PC-PAC

La conexión entre el ordenador y el autómatas se realiza a través del protocolo TCP (Transmission Control Protocol). Este protocolo se aplica mediante IP (Protocolo de Internet) a través de un cable Ethernet. El sistema IP establece direcciones que identifican a cada dispositivo para el intercambio de datos. En este caso, la dirección IP empleada para la transmisión de datos es v4.

Para la comunicación entre ambos dispositivos, se deben especificar las características de la red local y en el autómatas. La dirección IP del PC debe estar en el mismo rango que la dirección IP del autómatas. Esto significa que los tres primeros parámetros de la dirección deben coincidir, en este caso 192.168.3.X. El valor X es la dirección individual de cada equipo. El valor de la máscara subred debe permanecer en 255.255.255.0. Los valores de las IP's asignadas en el proyecto son los siguientes:

Dispositivo	IP	Máscara de subred
PC	192.168.3.249	255.255.255.0
PAC	192.168.3.10	255.255.255.0

Tabla 12: Configuración IP's

11.1.1. Configuración PC

Para configurar la IP del ordenador hay que dirigirse al Panel de Control y desde allí ir acceder al *Centro de redes y recursos compartidos* (1). Posteriormente hay que conectar el PC al autómatas de mediante el cable de red y el ordenador detectará automáticamente la conexión. Hacer clic en *Ethernet* (2).

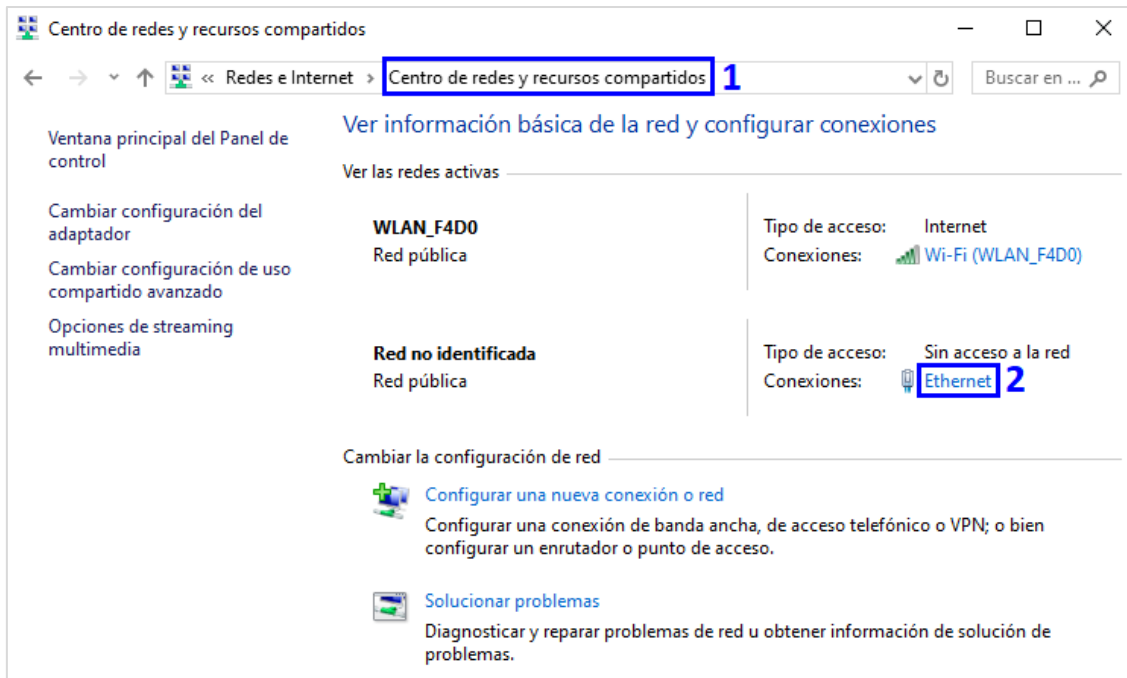


Figura 23: Configuración de la IP del PC

Seguidamente aparece la ventana *Estado de Ethernet* (3) y hay que hacer clic en *Propiedades* (4). A continuación, se abre una nueva ventana referente a las *Propiedades de Ethernet* (5), se selecciona el *Protocolo de Internet versión 4 (TCP/IPv4)* (6) y se clic de nuevo en *Propiedades* (7).

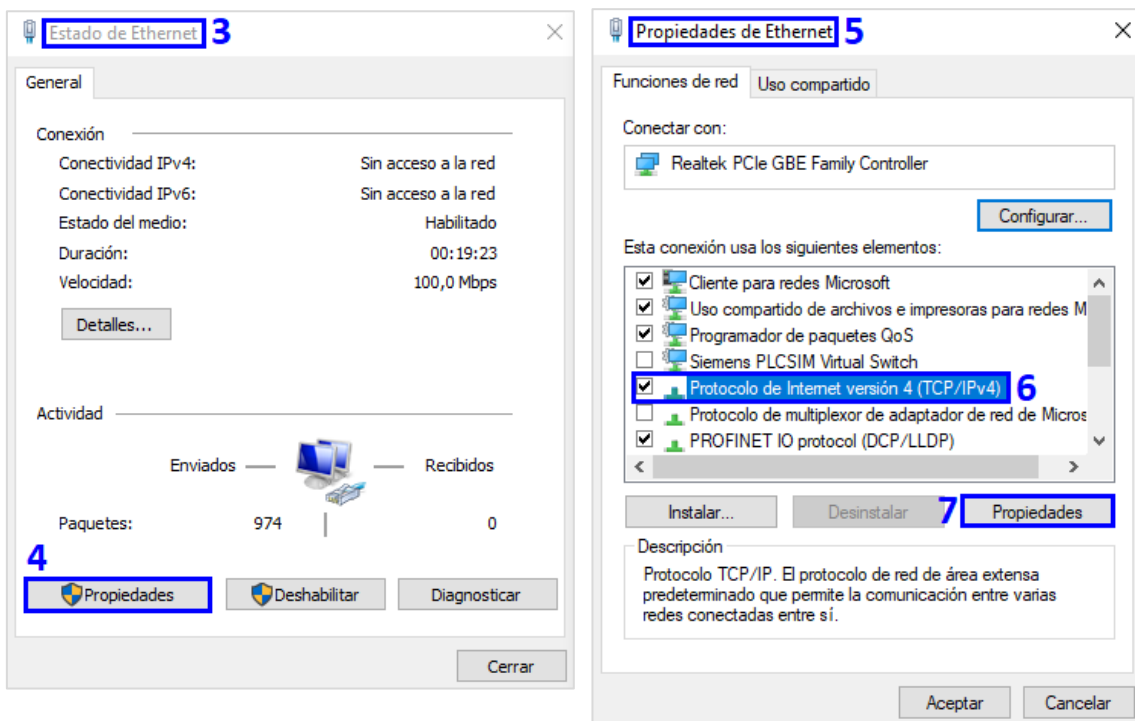


Figura 24: Configuración de la IP del PC.

Se muestra una pantalla donde se deben escribir los valores de IP (8) y máscara de subred (9) deseados. Posteriormente hay que hacer clic en aceptar para guardar los cambios introducidos.

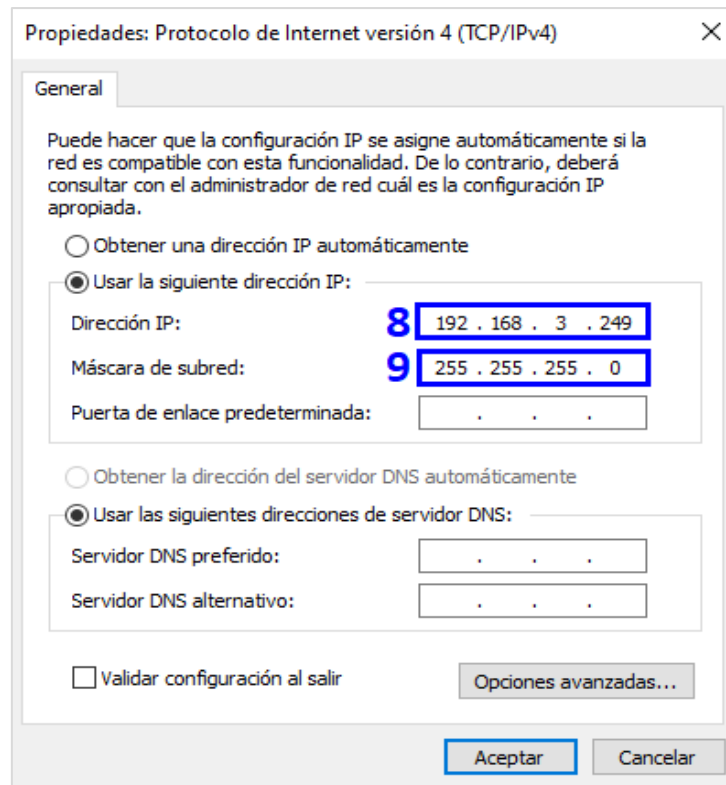


Figura 25: Configuración de la IP del PC

11.1.2. Configuración PAC

Para la configuración de la IP del autómat, primero hay que hacer doble clic en la pestaña de Ethernet (1) situada en el árbol del proyecto. Se abrirá el menú de Ethernet y posteriormente hay que seleccionar *Configuración IP* (2).

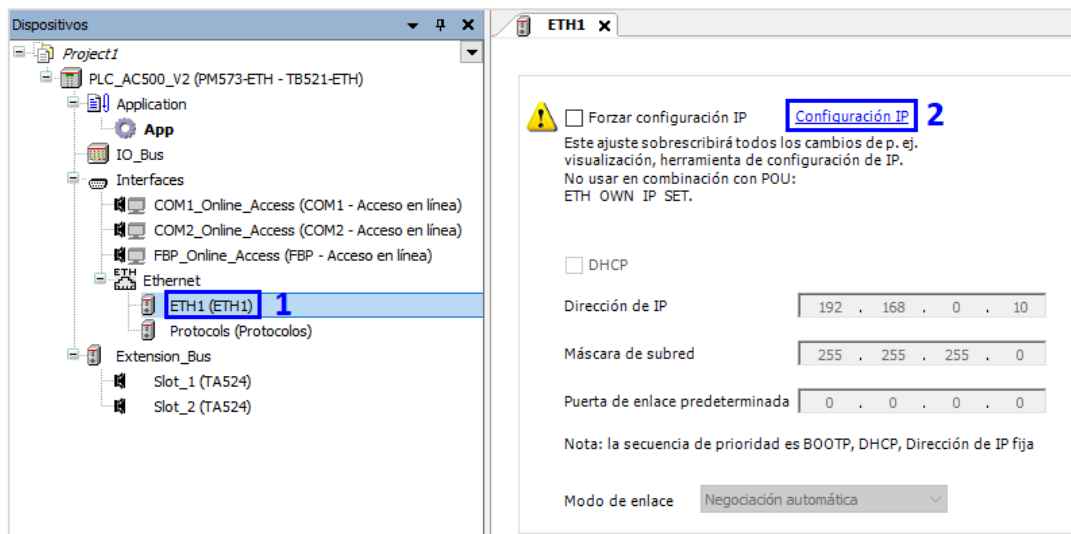


Figura 26: Configuración de la IP del PAC

A continuación se abrirá una nueva pestaña en el proyecto Configuración IP. Seleccionando el botón de *Scan* (3) se iniciará la búsqueda del autómatas (es necesario tener los equipos conectados). Una vez haya finalizado el proceso de búsqueda, se mostrará información relativa al autómatas entre la cual hay que destacar la dirección IP asignada en el proyecto al autómatas y la dirección IP configurada.

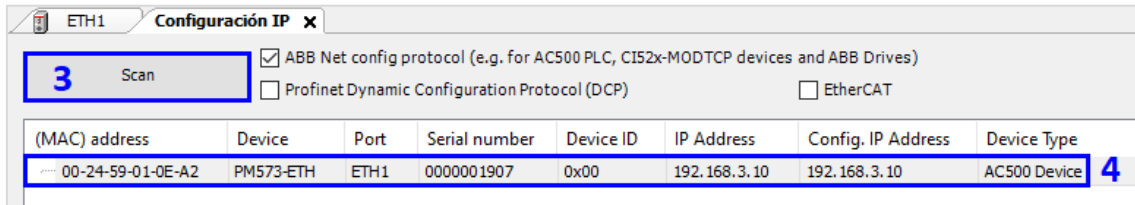


Figura 27: Configuración de la IP del PAC

Para modificar la dirección IP (5) y/o la máscara de subred (6) configurada en el autómatas hay que hacer clic en la información mostrada y en la parte inferior de la pestaña aparece un menú desde donde se gestiona. Para mandar la nueva configuración al autómatas (7) es necesario que autómatas no esté en modo run. Se puede modificar el estado del autómatas pulsando el botón físico RUN durante unos instantes. También se puede comprobar si la conexión se ha realizado con éxito utilizando la función de parpadeo (8).

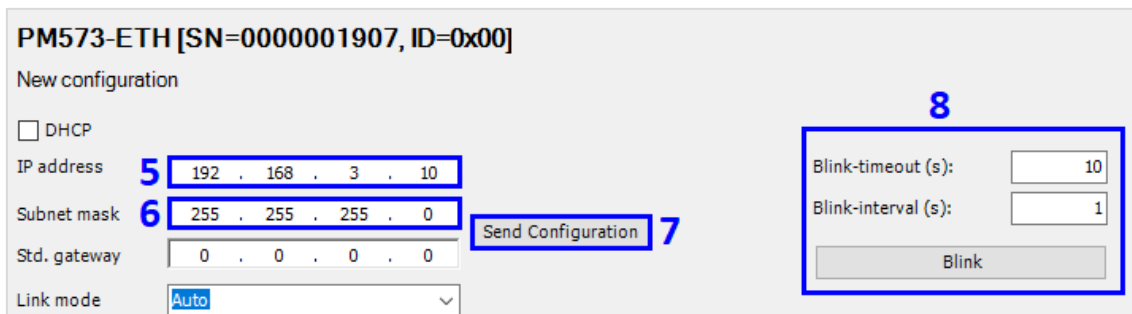


Figura 28: Configuración de la IP del PAC

11.2. Comunicación PAC-Convertidor

La conexión entre el autómatas y el convertidor se realiza a través del puerto serie (COM1) que es un bloque de terminales del autómatas mediante el protocolo RS-485. Este enlace, dispone de una arquitectura cliente-servidor. El variador de frecuencia se configura como servidor y el autómatas como cliente, puesto que será este quien realizará las peticiones de datos y el variador quien le de respuesta. La conexión se realiza mediante un bus lineal con transmisión half dúplex y con resistencia final de línea, la cual, impide que la señal refleje. El bus de datos es un cable de par trenzado apantallado que permite velocidades de transmisión de datos de entre 9,6 kbps y 12 Mbps.

11.2.1. Configuración PAC

Para realizar la configuración del autómatas programable mediante protocolo Modbus RS-485 hay que añadir el protocolo en el menú en forma de árbol. Para ello hay que hacer clic con el botón derecho del ratón la pestaña del COM1 (1) dentro de Interfaces y seleccionar *Agregar objeto* (2). A continuación se abre una ventana, hay que seleccionar *Protocolos serie* (3) y *COM1 - Modbus* (4) para añadir el protocolo al COM1. Después seleccionar *Sustituir objeto* (5) para realizar el cambio.

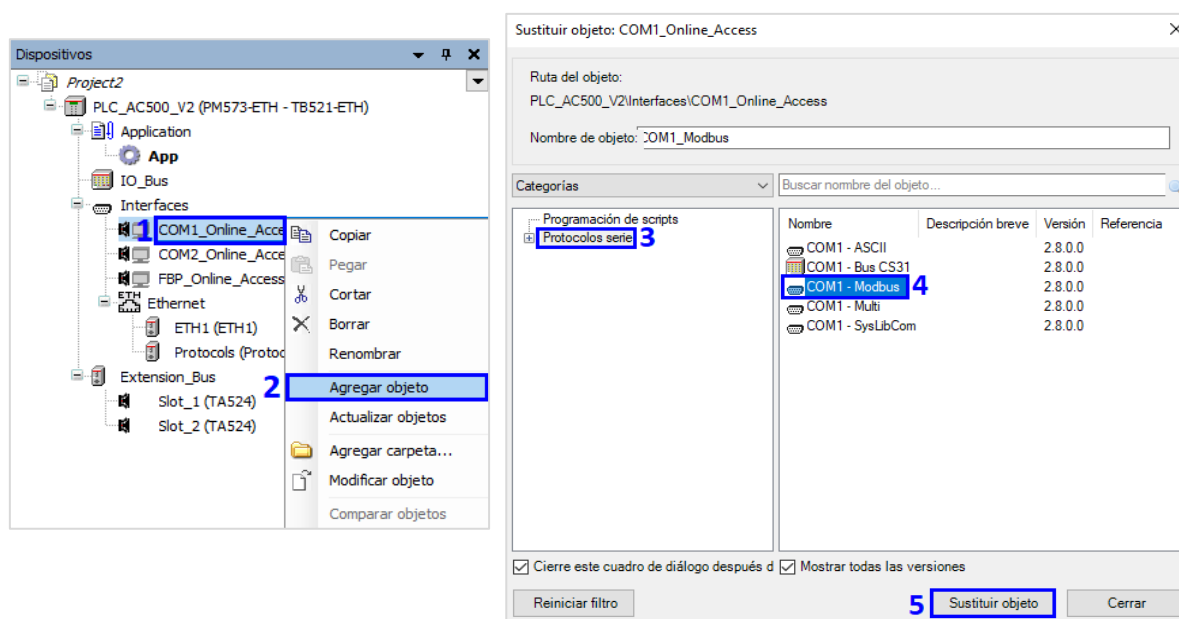


Figura 29: Configuración en RS-485 del PAC

Posteriormente hay que hacer doble clic en el COM1_Modbus y se despliega el menú de configuración del protocolo. De los parámetros que se muestran, se ha modificado el *Mando RTS* (6) que se tiene que ajustar en modo Telegrama para RS-485, se ha anulado el bit de paridad (8), se ha seleccionado el *Modo de Operación* (9) Cliente para el autómatas y por lo tanto la *Dirección* (10) debe ser 0.

COM1 - Modbus Configuración			
Configuración de servidor Modbus			
Parámetro	Tipo	Valor	
Habilitar inicio de sesión	Enumeration of BYTE	Deshabilitado	
6 Mando RTS	Enumeration of BYTE	Telegrama	
Valor de fin del telegrama	WORD(0..65535)	3	
Velocidad en baudios	Enumeration of DWORD	19200	
7 Paridad	Enumeration of BYTE	Ninguno	
Bits de datos	Enumeration of BYTE	8	
Bits de paro	Enumeration of BYTE	1	
Ejecutar con error de configuración	Enumeration of BYTE	No	
8 Modo de operación	Enumeration of BYTE	Cliente	
9 Dirección	BYTE(0..255)	0	

Figura 30: Configuración en RS-485 del PAC

En cuanto a la Configuración del servidor Modbus destacar que no es necesario realizar cambios. Se debe usar la zona de memoria %M y no la de registrados %R tal y como está predeterminado en el software.

11.2.2. Configuración convertidor

Para configurar el convertidor para la comunicación mediante el bus de campo integrado es necesario indicar las propiedades de la comunicación y ajustar los parámetros del grupo 58.

Parámetro	Ajuste seleccionado para el control de bus de campo	Función/Información
58.01 Habilitar protocolo	Modbus RTU	Habilita/deshabilita la interfaz de bus de campo integrada y selecciona el protocolo que se debe usar.
58.03 Nodo	Nodo 1	Define el nodo del convertidor en el enlace de bus de campo. Están permitidos los valores 1 a 247.
58.04 Velocidad transmisión	19,2 Kb/s	Selecciona la velocidad de transferencia del enlace de bus de campo.
58.05 Paridad	Ninguno 1 (8 bits de datos, sin bit de paridad y un bit de paro)	Selecciona el ajuste para la paridad y el bit de paro. Hay que utilizar el mismo ajuste que en la estación maestra.
58.06 Control comunicación	Habilitado	Funcionamiento normal. Asume los ajustes del BCI cambiados durante el funcionamiento.
58.14 Pérdida comunicación	Fallo	Define la medida que se toma cuando se detecta una pérdida de comunicación.
58.33 Modo direccionamiento	Modo 0	Define el mapeo entre parámetros y registros de retención en el rango de registros de Modbus.

Tabla 13: Configuración BCI convertidor

11.3. Comunicación PAC-HMI Modbus RS-485

La conexión entre el PAC y el panel de control se realiza mediante el protocolo Modbus RS-485 del conector COM 2 del autómatas y el puerto del HMI. En este caso, en la arquitectura del sistema, el autómatas se configura como servidor y el HMI como cliente (el autómatas servirá las peticiones de datos del HMI). La conexión se realiza con un bus de datos de par trenzado y apantallado y también dispone de resistencia de final de línea. La velocidad de transmisión de datos se mantiene en 19,2 Kb/s.

11.3.1. Configuración PAC

La configuración mediante RS-485 del autómatas para las comunicaciones con el HMI es similar a la configurada en el caso del convertidor de frecuencia. El primer paso es ir al árbol del proyecto, hacer clic con el botón derecho en el COM2 y seleccionar *Agregar objeto*. Posteriormente seleccionar *COM 2 – Modbus* y *Sustituir objeto* para proceder a realizar los cambios.

A continuación, hay que hacer doble clic sobre la pestaña de COM 2 y se abre una ventana con la configuración Modbus del COM 2. Se ha configurado el *Mando RTS* (1) en modo Telegrama para comunicaciones mediante RS-485, sin bit de paridad (2), y en este caso que el PAC sirve la información al HMI este se configura como servidor y se le asigna el nodo de dirección 1.

COM2 - Modbus Configuración			
Configuración de servidor Modbus			
Parámetro	Tipo	Valor	
Habilitar inicio de sesión	Enumeration of BYTE	Deshabilitado	
1 Mando RTS	Enumeration of BYTE	Telegrama	
Valor de fin del telegrama	WORD(0..65535)	3	
Velocidad en baudios	Enumeration of DWORD	19200	
2 Paridad	Enumeration of BYTE	Ninguno	
Bits de datos	Enumeration of BYTE	8	
Bits de paro	Enumeration of BYTE	1	
Ejecutar con error de configuración	Enumeration of BYTE	Sí	
3 Modo de operación	Enumeration of BYTE	Servidor	
4 Dirección	BYTE(0..255)	1	

Figura 31: Configuración en RS-485 del PAC

11.3.2. Configuración HMI

Para configurar un proyecto de Panel Builder 600 hay que ir al menú del árbol del proyecto, y dentro de la pestaña *Config* hacer doble clic en *Protocols* (1). Tras ello se abrirá una pestaña a través de la cual se puede seleccionar un protocolo y configurarlo. Para seleccionar el protocolo hay que hacer clic en la columna con la etiqueta *PLC* (2) y se abrirá un desplegable con todas las opciones de los protocolos disponibles para implementar. En este caso se ha seleccionado ABB Modbus RTU. Para su configuración hay que hacer clic en la siguiente columna de *Configuración* (3) y se abrirá una nueva ventana de diálogo.

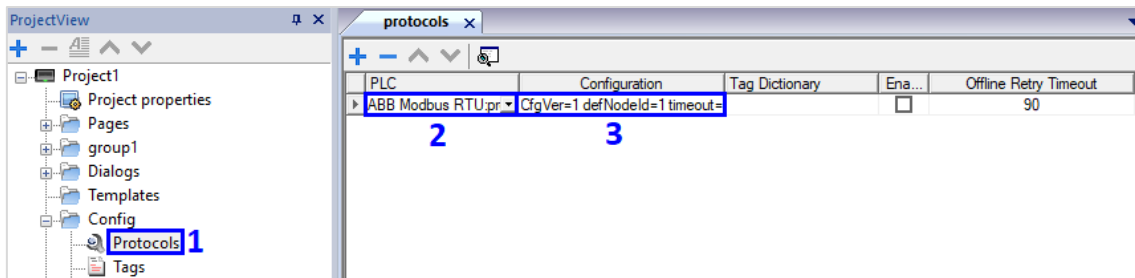


Figura 32: Configuración en RS-485 del HMI

La nueva ventana primeramente hay que configurar el *Node ID* (4). Este parámetro debe coincidir con el nodo configurado del COM2 del autómatas. A continuación hay que escoger el *Transmission Mode* (5) y el *modelo de PLC* (6). El modo de transmisión se ha seleccionado RTU puesto que el protocolo escogido es Modbus RTU en RS-485 y el modelo del PLC empleado en el proyecto es un PM573-ETH. Para continuar configurando más opciones hay que seleccionar el botón *Com...* (7). Al hacer clic se muestra una nueva ventana con las opciones más específicas de Modbus. Para el puerto del HMI se ha seleccionado el *com1* (8) puesto que este solo dispone de un puerto. En cuanto a la velocidad de transmisión, la paridad, la longitud de bits y la cantidad de bits de paro (9) se les han atribuido los mismos valores que en el PAC puesto que estos deben ser coincidentes. El modo de transmisión es *RS-485* (10). Para finalizar seleccionar *OK* (11) en ambas ventanas para guardar los cambios introducidos.

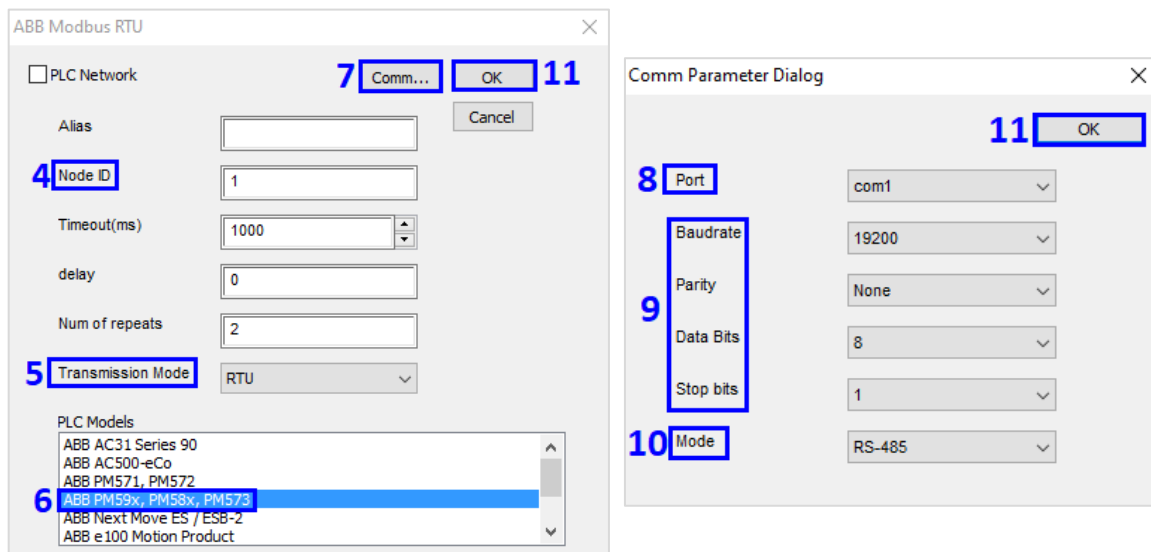


Figura 33: Configuración en RS-485 del HMI

11.4. Comunicación PAC-HMI Modbus TCP

La comunicación entre el autómatas y el panel de control también se ha implementado mediante el protocolo Modbus TCP. Pese a que no se pueden utilizar dos protocolos simultáneamente para una misma conexión se ha configurado la conexión también en Modbus TCP para descartar posibles fuentes de errores en el transcurso de las comunicaciones. La configuración de las comunicaciones es similar la realizada en RS-485 y el cable de conexión entre equipos ha sido el mismo.

11.4.1. Configuración PAC

Para la configuración del autómatas en el protocolo Modbus TCP, hay que ir al árbol del proyecto y hacer clic con el botón derecho en *Protocols* (1) y después seleccionar *Agregar objeto* (2). A continuación se abrirá una nueva ventana y al marcar *Protocolos Ethernet* (3) se mostrarán los protocolos disponibles. La opción para esta configuración es *Servidor TCP/IP Modbus* (4). Para salir presionar *Agregar objeto* (5).

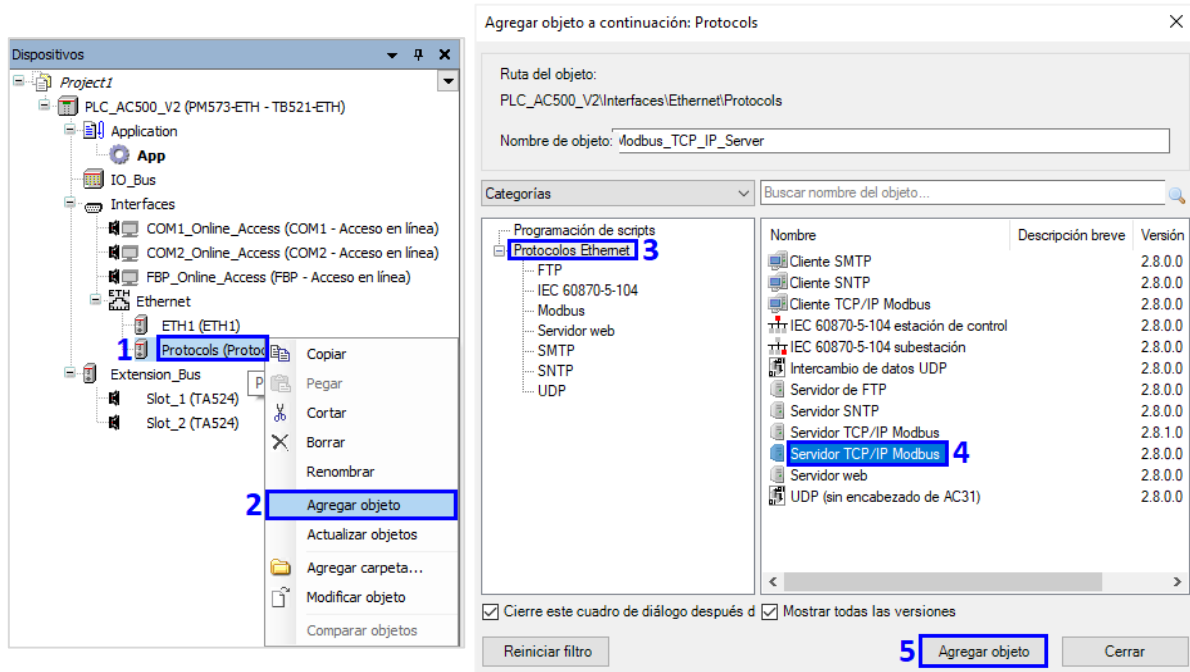


Figura 34: Configuración en TCP/IP del PAC

Una vez se ha insertado el protocolo en el proyecto hay que hacer doble clic en el protocolo (6) para abrir una pestaña donde se muestran sus opciones de configuración. De las opciones disponibles únicamente hay que modificar el número de *Conexiones del servidor* (7). El servidor Modbus debe usar la zona de memoria %M tal y como está predeterminado.

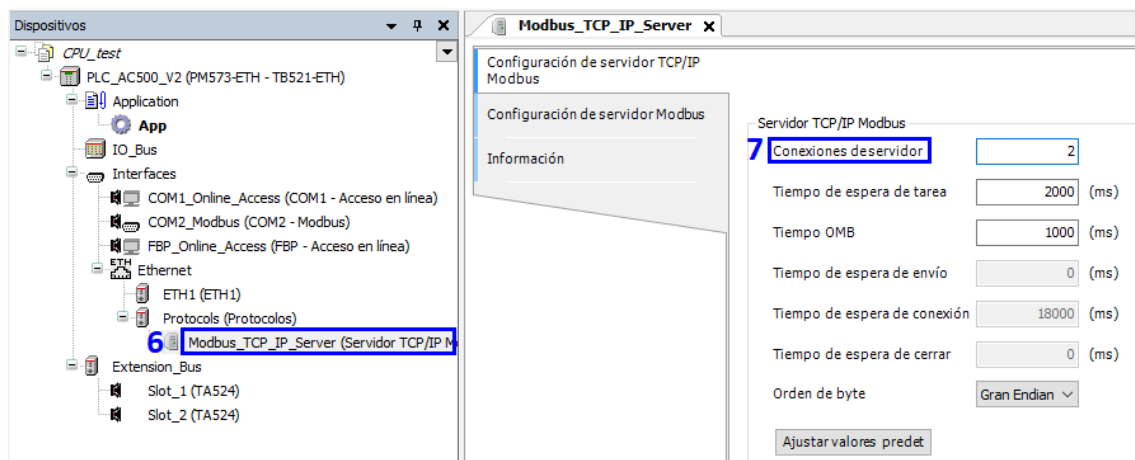


Figura 35: Configuración en TCP/IP del PAC

11.4.2. Configuración HMI

Para constituir la configuración en el panel de control es necesario hacer doble clic en la pestaña de *Protocols* (1) en el menú en forma de árbol del proyecto. En la columna del PLC se debe abrir el desplegable y seleccionar *ABB Modbus TCP* (2). Seguidamente hay que hacer clic en la pestaña de *Configuración* (3).

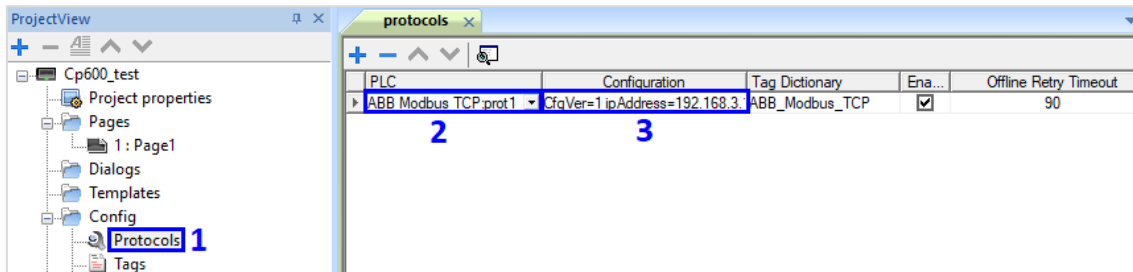


Figura 36: Configuración en TCP/IP del HMI

A continuación aparecerá una nueva ventana en la cual hay que configurar la *dirección IP* (4) e introducir el *modelo de PLC* (5) empelado en el proyecto. La dirección que hay que introducir es la correspondiente al autómatas y el modelo de autómatas es el PM573-ETH. En cuanto al puerto, hay que seleccionar el 502 por que este protocolo se conecta predeterminadamente a este puerto. Por último seleccionar *OK* (6) para guardar la configuración establecida.

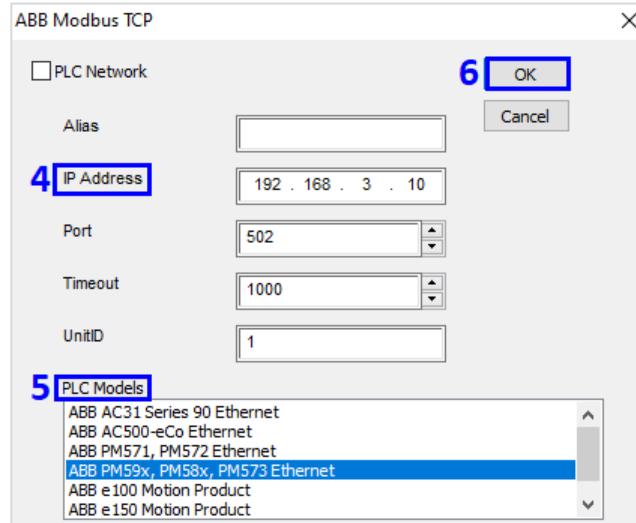


Figura 37: Configuración en TCP/IP del HMI

12. CONTROL Y SUPERVISIÓN DEL MOTOR

El sistema desarrollado permite controlar y supervisar el funcionamiento de un motor trifásico. Este proceso sobre el motor se puede realizar mediante el HMI o la maqueta en la que está instalado el variador mediante la conmutación de un interruptor (EXT1/EXT2) disponible en el panel de control, se puede elegir al fuente de control. Por defecto el control se realiza desde la maqueta. Ambos dispositivos tienen capacidad de lectura y escritura sobre las variables seleccionadas, sin embargo la maqueta actúa sobre las entradas y salidas físicas y el HMI lo hace sobre la memoria del variador. En este sistema el autómatas actúa de enlace entre el HMI y el convertidor de frecuencia, es decir, interpreta la información y realiza la transferencia de datos entre ambos además de implementar algunas funciones para el control.

12.1. Programación del convertidor

La programación del convertidor consiste básicamente en configurar los parámetros en función de las necesidades del usuario. A lo largo del proyecto se han realizado diferentes fases de configuración. En primera instancia, se han introducido los valores nominales característicos del motor en el variador, por tal de realizar la marcha de identificación y que el variador estime los valores reales mediante un modelo físico. La segunda fase de configuración radica en la selección del protocolo de comunicación y los parámetros que lo componen por tal de establecer correctamente la conexión con el autómatas. La última fase ha consistido en configurar los parámetros relacionados con las entradas y salidas digitales y analógicas del variador para el control y supervisión de este sobre la maqueta y la configuración de los parámetros de funcionamiento del convertidor necesarios para realizar el control y supervisión desde el HMI. En los anexos se dispone de una tabla donde se muestran todos los parámetros configurados del convertidor de frecuencia.

12.2. Programación del autómatas

Como se ha comentado anteriormente, la función del autómatas (PAC Programmable Automation Controller) en este sistema consiste en enlazar el convertidor y el panel de control y permitir la transferencia de datos entre ambos dispositivos además de implementar las funciones para el control. Por lo tanto su programación se enfoca básicamente a estos aspectos. Para su implementación se han usado los lenguajes de programación Structured Text (ST), Ladder (LD) y Function Block Diagram (FBD) en función de las necesidades. Para la realización de diagramas de contactos simples se ha utilizado Ladder, en las funciones más complejas se ha empleado Structured Text y para la creación de bloques de funciones se ha utilizado FBD. A continuación se muestran los programas desarrollados para llevar cabo estas funciones.

12.2.1. Programa PLC_PRG

Este programa es la unidad principal de programación. Su función es invocar al resto de subprogramas para que estos se ejecuten en el módulo.

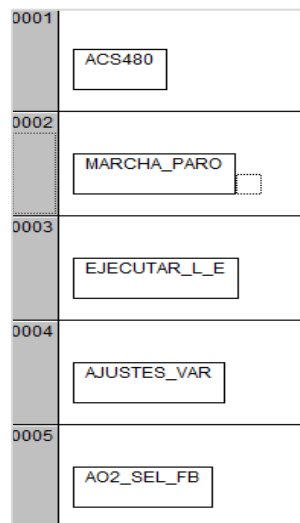


Figura 38: Programa PLC_PRG

12.2.2. Programa ACS480

Este programa permite las comunicaciones entre el convertidor de frecuencia ACS480 y el autómat. Para su realización se ha descargado la librería PS553-DRIVES de la página web oficial de ABB, la cual, es una librería únicamente para productos de ABB que permite la comunicación directa mediante diferentes protocolos, el control básico y acceso a parámetros del variador. En las entradas de estos bloques es necesario introducir las variables o señales para el funcionamiento del bloque, y en las salidas se puede observar el funcionamiento del bloque. A continuación se muestran y describen los diferentes bloques de funciones empleados en este programa.

12.2.2.1. Bloque de funciones ACS_COM_MOD_RTU

Este bloque funcional permite realizar un control básico de las comunicaciones mediante el protocolo Modbus RTU con los dispositivos de las gamas ACS y DCS. En este caso permite la comunicación entre el ACS480 y el AC500.

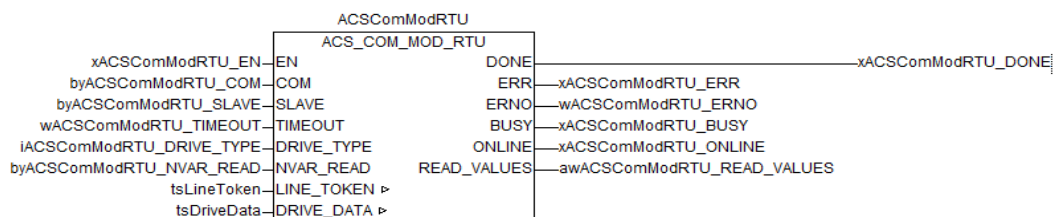


Figura 39: Bloque ACS_COM_MOD_RTU

12.2.2.2. Bloque de funciones ACS_DRIVES_CTRL_STANDARD

Este bloque de funciones se usa para controlar convertidores de la gama ACS con dispositivos de la gama ABB. Este proporciona señales de marcha y paro a la unidad de control y diagnósticos de las señales leídas. En las entradas se le proporciona información relacionada con las señales básicas de funcionamiento y referencias, y en la salida se puede observar el estado del control.

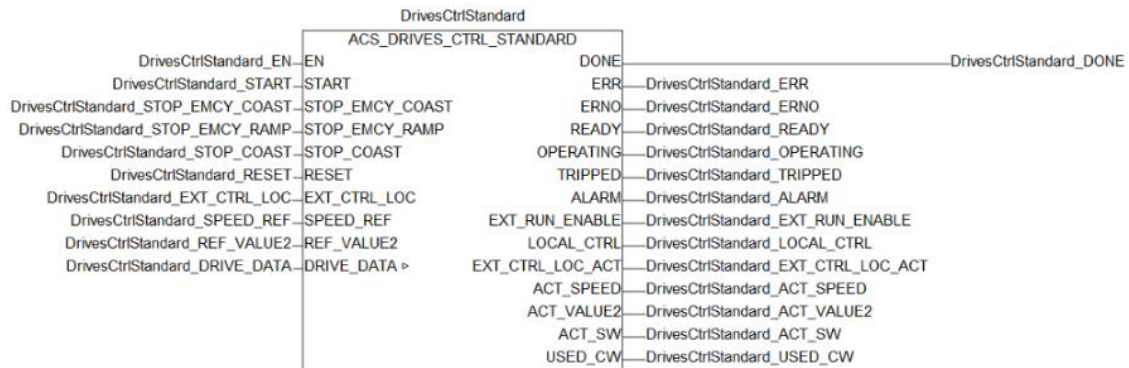


Figura 40: Bloque ACS_DRIVES_CTRL_STANDARD

12.2.2.3. Bloque de funciones ACS_MOD_READ_N_PRM

Este bloque permite la lectura de parámetros del convertidor. Para ello es necesario saber la dirección Modbus en la que se ha codificado el parámetro en cuestión y asociarle una variable de programa en la que se guarde la información.

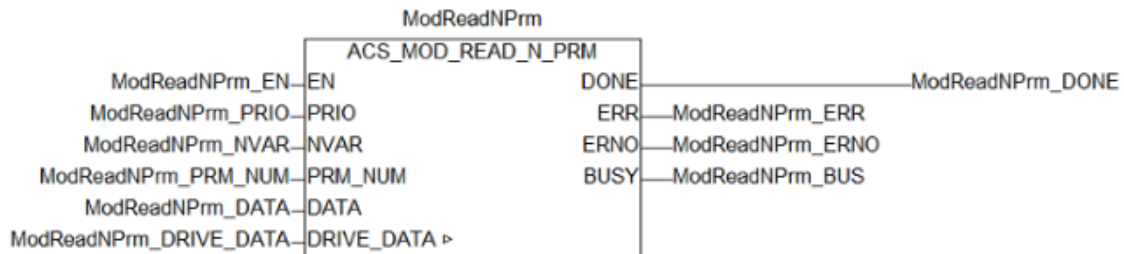


Figura 41: Bloque MOD_READ_N_PRM

12.2.2.4. Bloque de funciones ACS_MOD_WRITE_N_PRM

Este bloque permite la escritura de parámetros en el convertidor. Para ello es necesario saber la dirección Modbus en la que se ha codificado el parámetro en cuestión y e indicar la variable de la cual tiene que copiar el valor.

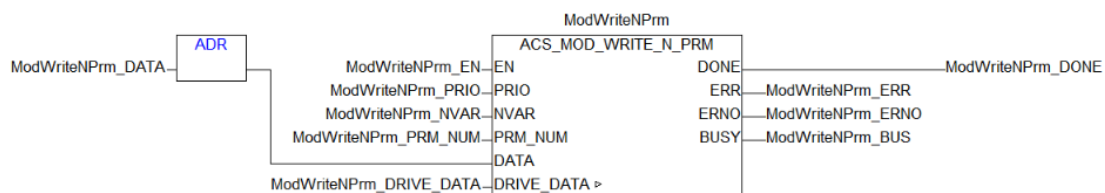


Figura 42: Bloque MOD_WRITE_N_PRM

12.2.2.5. Código programa ACS480

El código de este programa se encuentra en los anexos del documento.

12.2.3. Programa MARCHA_PARO

Cuando se habilita el control del motor desde el HMI (EXT2), este programa permite realizar el control de marcha y de paro del bloque de funciones ACS_DRIVES_CTRL_STANDARD que a su vez lo realiza sobre el variador y por lo tanto sobre el motor.

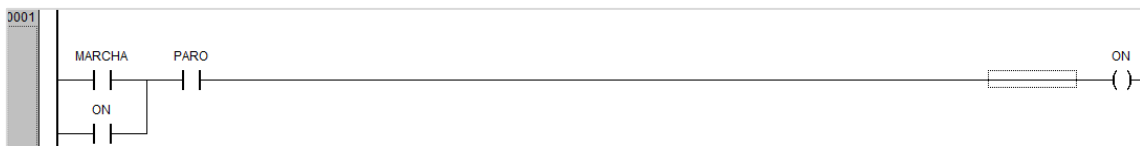


Figura 43: Programa MARCHA_PARO

12.2.4. Programa EJECUTAR_L_E

La finalidad de este programa es que no se puedan realizar simultáneamente las funciones de escritura y lectura. De este modo se pueden leer el estado de una variable en el convertidor y modificarlo desde una misma variable, sin tener que generar una variable para la escritura y otra para la lectura.

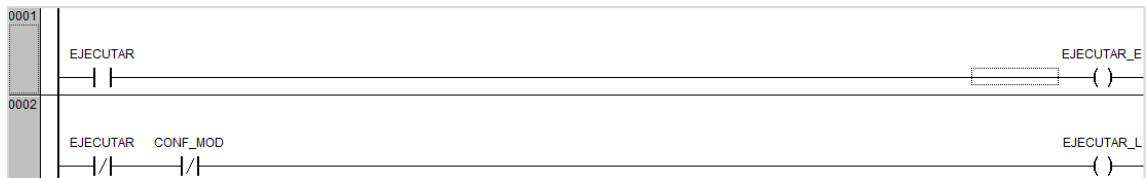


Figura 44: Programa EJECUTAR_L_E

12.2.5. VEL_SEL_FB

Este programa interpreta el valor de la posición del selector de velocidades del panel de control y lo modifica para proporcionar la referencia de velocidad en un formato que el bloque de control del motor sea capaz de ejecutar.

```

0001 IF V_SEL = 1 THEN
0002   VEL_REF2:=500;
0003 ELSIF V_SEL = 2 THEN
0004   VEL_REF2:=1000;
0005 ELSIF V_SEL = 3 THEN
0006   VEL_REF2:=1500;
0007 ELSIF V_SEL <1 OR V_SEL >3 THEN
0008   VEL_REF2:=500;
0009 END_IF
0010

```

Figura 45: Programa VEL_SEL_FB

12.2.6. AO2_SEL_PRG

Este programa interpreta el valor seleccionado del menú de selección de salidas analógicas 2 del panel de control y hace la conversión de este valor para que el convertidor muestre la variable seleccionada.

```

0001 IF AO2_SEL = 0 THEN
0002 AO2_FUENTE:=5;
0003
0004 ELSIF AO2_SEL = 1 THEN
0005 AO2_FUENTE:=27;
0006
0007 ELSIF AO2_SEL = 2 THEN
0008 AO2_FUENTE:=8;
0009
0010 ELSIF AO2_SEL = 3 THEN
0011 AO2_FUENTE:=6;
0012
0013 END_IF

```

Figura 46: Programa AO2_SEL_PRG

12.2.7. AJUSTES_VAR

El objetivo de este programa es ajustar los valores de diversas variables. En primer lugar en función del parámetro de velocidad leído del convertidor se determina el sentido de giro del motor. En el segundo, se realiza un ajuste en estas variables en función si son de lectura o escritura puesto que el escalado es diferente en el variador que en el panel de control. En el tercer subprograma se realiza un ajuste del parámetro seleccionado mediante el interruptor para el canal de control. Este se convierte un valor booleano que el variador puede interpretar.

```

0001 (*DIRECCION*)
0002 IF VELOCIDAD >=0 THEN
0003     DIRECCION:=FALSE;
0004 ELSIF VELOCIDAD <0 THEN
0005     DIRECCION:=TRUE;
0006 END_IF
0007
0008 (*AJUSTE VARIABLES LECTURA Y ESCRITURA*)
0009 IF EJECUTAR=TRUE THEN
0010     PAR_MIN_100_E:=PAR_MIN_100_L;
0011     PAR_MAX_100_E:=PAR_MAX_100_L;
0012     T_ACC_E:=T_ACC_L;
0013     T_DEC_E:=T_DEC_L;
0014 END_IF
0015
0016 (*AJUSTE SELECCIÓN CANAL DE CONTROL EXTERNA*)
0017 IF EXT_SEL = 1 THEN
0018     EXT_SEL_2:=FALSE;
0019 ELSIF EXT_SEL = 2 THEN
0020     EXT_SEL_2:=TRUE;
0021 END_IF

```

Figura 47: Programa AJUSTES_VAR

12.3. Programación HMI

El sistema HMI permite de manera intuitiva realizar un control sobre el motor y supervisar los parámetros de funcionamiento. En la pantalla inicial se muestran los elementos de control esenciales como son los pulsadores de marcha y de paro, la selección de la fuente de control, y la barra de selección de velocidades constantes. En cuanto a los indicadores se puede visualizar el estado del convertidor, la dirección del motor y los parámetros de velocidad e intensidad. En la parte superior se encuentra cuatro botones que permiten la navegación por las diferentes pantallas.



Figura 48: Pantalla de Inicio

En Parámetros se accede a la visualización de los parámetros del convertidor de frecuencia. Se han seleccionado las variables consideradas clave para la supervisión de un motor. Estas variables son únicamente de lectura.



Figura 49: Pantalla de Parámetros

El apartado de Gráficos contiene diversas pantallas a través de las cuáles se puede navegar mediante los botones que contienen las flechas. Cada pantalla está compuesta por un gráfico en el que se represan diferentes parámetros en función del tiempo.

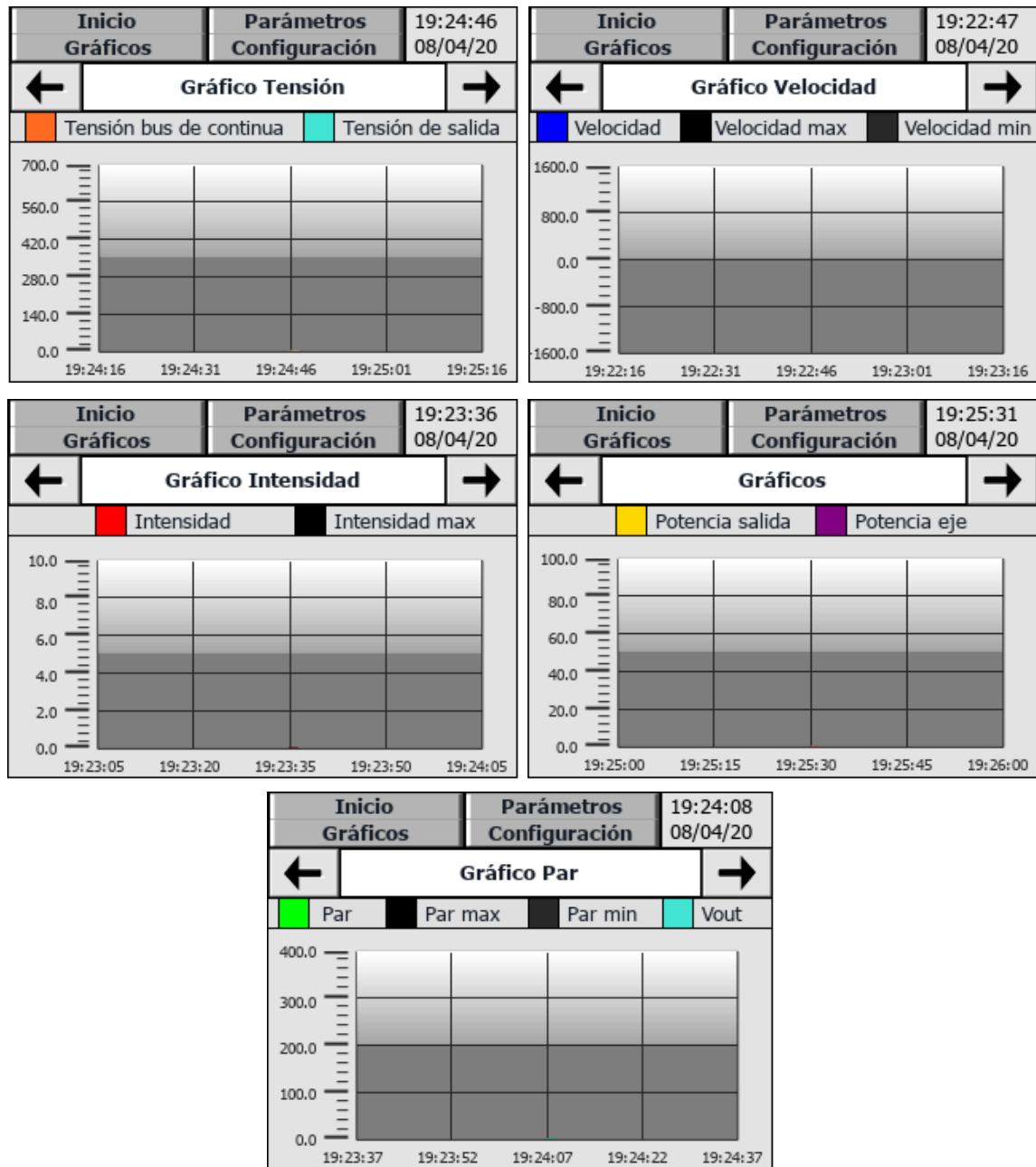


Figura 50: Pantallas Gráficos

El apartado de Configuración permite la modificación de los máximos y los mínimos de la velocidad y el par, la modificación de las rampas de aceleración y deceleración y la seleccionar la variable analógica que se desea visualizar en la maqueta. Al seleccionar el botón de Configuración se abre un cuadro de diálogo que informa sobre la necesidad de tener el motor en estado de paro y de acceder al usuario admin por tal que el programa permita modificar los parámetros.

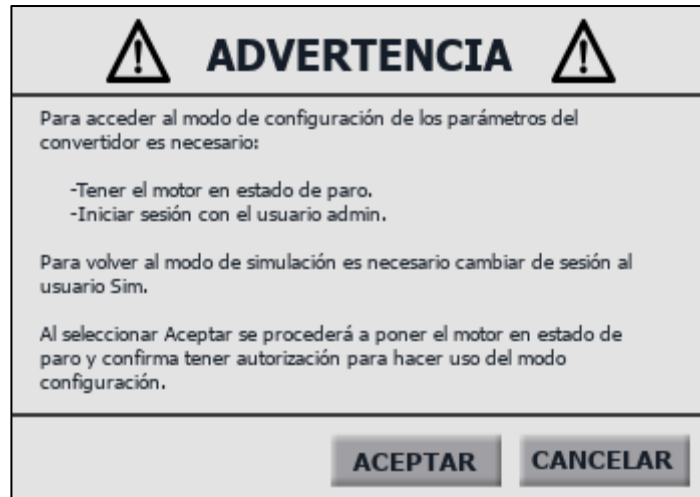


Figura 51: Cuadro diálogo Configuración.

En el caso de seleccionar Aceptar el programa da orden de paro al motor y permite el acceso al apartado de Configuración en modo lectura, en caso de seleccionar Cancelar se cierra el cuadro de diálogo y se retorna a la página anterior. En las pantallas de configuración es necesario pulsar el botón aplicar por tal de escribir el convertidor de frecuencia los cambios introducidos.

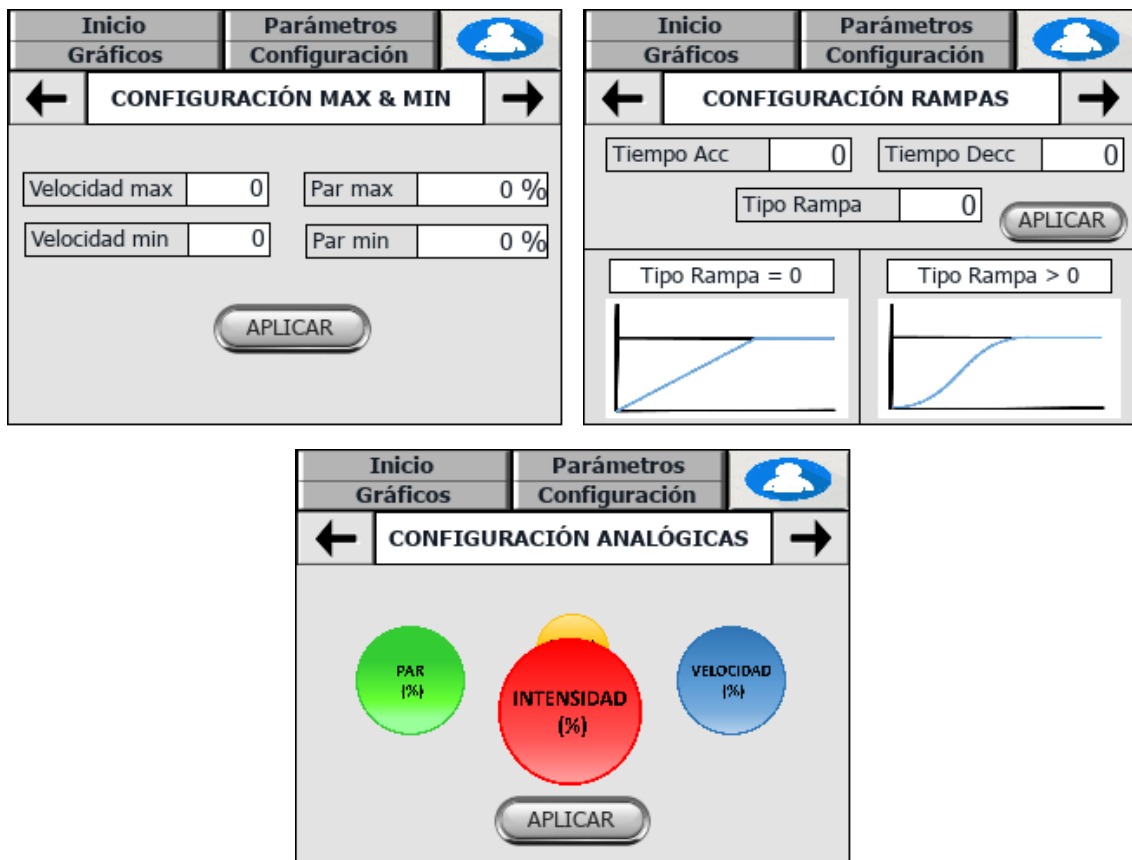


Figura 52: Pantallas Configuración

El sistema dispone de dos usuarios con permisos diferentes. Es posible realizar el cambio de usuario haciendo clic en el símbolo situado en el extremo superior derecho. El usuario Sim y sin contraseña dispone de permiso totales de navegación y lectura. También tiene permisos para modificar el estado de los pulsadores y selectores de la pantalla de inicio, pero no puede modificar los parámetros de configuración. El usuario Admin con contraseña 123 únicamente tiene permisos de navegación, de escritura y lectura por las pantallas del modo configuración. Este sistema de usuarios garantiza que mientras se está operando en el modo configuración, el motor está parado y no se puede poner en marcha.



Figura 53: Pantalla cambio usuario

12.4. Funcionalidad de la Maqueta

Las funcionalidades de la maqueta se han programado a través de la configuración de los parámetros del convertidor de frecuencia. Las funciones de las entradas digitales DI1, DI2 y DI3 son las de Marcha, Paro y selección de la Dirección del motor.

Estado DI1 (Marcha)	Estado DI2 (Paro)	Estado DI3 (Dirección)	Comando
0→1	1	0	Marcha en avance
0→1	1	1	Marcha en retroceso
Cualquiera	0	Cualquiera	Paro

Tabla 14: Funciones DI1, DI2 y DI3

Las entradas digitales DI4 y DI5 se utilizan para la selección de velocidades. En caso de que la velocidad seleccionada sea 00 la referencia de velocidad se gestiona mediante el potenciómetro de la entrada analógica AI1.

Estado DI4	Estado DI5	Velocidad
0	0	Regulación mediante potenciómetro AI1
0	1	500 rpm
1	0	1000 rpm
1	1	1500 rpm

Tabla 15: Funciones DI4 y DI5

La entrada digital DI6 permite cambiar entre las dos series de tiempos de rampa de aceleración/deceleración definidos en el convertidor. Los tiempos de las rampas 1, son modificables mediante el panel de control.

Estado DI6	Parámetro	Tiempo
0	Rampa 1 aceleración	20 s
	Rampa 1 deceleración	20 s
1	Rampa 2 aceleración	10 s
	Rampa 2 deceleración	10 s

Tabla 16: Función DI6

Las entradas analógicas permiten en control de la referencia de velocidad. En el caso de la entrada analógica AI1 esta consigna se proporciona mediante el potenciómetro físico del panel. En el caso de la AI2 esta consigna se proporciona desde el panel de control y por lo tanto la fuente de selección es el bus de campo integrado.

En el caso de las salidas analógicas, su función es mostrar parámetros del convertidor. En la AO1 se muestra la frecuencia del convertidor, y en la AO2 predeterminadamente se muestra la intensidad del motor en tanto por ciento de la intensidad nominal, pero también se pueden seleccionar, el par motor en tanto por ciento, la potencia de salida en kW y la velocidad del motor absoluta en tanto por ciento.

12.5. Motor seleccionado para las pruebas

El motor con el que se han llevado a cabo todas las pruebas del presente proyecto es un motor trifásico de tipo asíncrono de la marca ABB. El control realizado sobre el motor ha sido de tipo vectorial.



Figura 54: Motor ABB 4kW

Parámetros nominales motor ABB	
Tensión nominal motor	400 V
Frecuencia nominal	50 Hz
Velocidad nominal	1430 rpm
Pares de polos motor	2
Potencia nominal	4 kW
Cos ϕ nominal	0.82
Intensidad nominal	8,72 A
Grado de protección	IEC34-1
Peso	45 Kg

Tabla 17: Parámetros nominales del motor

13. CONCLUSIONES

Una vez finalizado el proyecto se puede concluir que se han cumplido los objetivos propuestos con éxito. Se ha logrado la interconexión de los diferentes equipos industriales de modo, que se ha creado una plataforma de control y supervisión industrial robusta y fiable.

Los equipos y software empelados disponen de gran cantidad de documentación para iniciarse en sus herramientas y muestran un buen rendimiento. Entre los software destacar Codesys, el cual, es gratuito, permite gran flexibilidad y se fundamenta en la norma IEC 61131. También muestra un gran potencial para el desarrollo de aplicaciones complejas, puesto que dispone de distintos lenguajes de programación y permite la creación de SCADA.

De cara al futuro, existe la posibilidad de realizar numerosas ampliaciones de este proyecto. Por ejemplo, se podría utilizar el módulo de expansión del autómatas de encoder, contador y señales PWM, lo que permitiría realizar el control a tiempo real de servoaccionamientos. También existe la posibilidad de realizar las comunicaciones entre equipos mediante otros protocolos de comunicaciones, lo cual, permitiría el desarrollo de aplicaciones guardarán los históricos de información en la nube o, enviarán mensajes al correo o a dispositivos móviles en caso de alarma o emergencia.

En el transcurso del proyecto se han encontrado diferentes dificultades técnicas, las cuáles, se han solventado a excepción de una. Al poner en marcha el sistema, el autómatas y el panel de control no establecen comunicaciones de forma predeterminada. Para iniciar la comunicación entre ambos dispositivos es necesario conectar el ordenador al autómatas e iniciar sesión en él. Por tal de solucionar este error se han realizado múltiples acciones, entre ellas destacar que se ha establecido las comunicación mediante otro protocolo (Modbus TCP), se ha contactado con el servicio técnico de ABB, al cual, se le ha enviado los programas realizados para la ejecución de este proyecto por tal de que ellos realicen las simulaciones con sus equipos, y se han realizado pruebas con una pantalla nueva. Sin embargo ninguna de ellas ha dado solución al problema. Se tenían planteadas otras acciones como la solicitud de una licencia Premium o solicitar asistencia remota, pero con la irrupción de la pandemia se han tenido que descartar.

En la figura 55 se muestra el diagrama de Gantt de la evolución del proyecto. Como se puede observar, la programación inicial se ha visto modificada principalmente por el inicio de mi vida laboral y unos problemas técnicos con la CP630.

A título personal, estoy satisfecho con el resultado final. El proyecto me ha permitido consolidar los conocimientos adquiridos a lo largo de estos años de estudio y profundizar en equipos y tecnologías ampliamente extendidos en el mundo de la automatización industrial. De este modo he adquirido competencias universales en la materia, y por lo tanto, he aumentado mis capacidades más allá de las restricciones que proporcionan los equipos de un mismo fabricante que seguro me serán útiles en mi futuro profesional.

	Año 2019									Año 2020			
Acciones	Abr	May	Jun	Jul	Ago	Set	Oct	Nov	Dic	Ene	Feb	Mar	Abr
Recopilación información general													
Descarga e instalación software													
Cableado variador-maqueta													
Cableado del autómeta													
Configuración básica convertidor													
Comunicación PC-PAC													
Pruebas variador-maqueta													
Comunicación PAC-HMI (RS-485)													
Programación interfaz HMI													
Programación PAC-Convertidor													
Redacción del documento													
Configuración convertidor													
Comunicación PAC-HMI (TCP)													
Reescalado variables convertidor													
Pruebas de conjunto													

Figura 55: Evolución del proyecto

14. BIBLIOGRAFIA

- ABB Library. (2018). *Familia de productos AC500*. Recuperado de <https://library.abb.com/es>
- ABB Library. (2016). *Plataforma de automatización AC500. Herramienta de ingeniería Automation Builder 1.2*. Recuperado de <https://library.abb.com/es>
- ABB Library. (2009). *Plataforma de automatización escalable AC500. Modbus RTU*. Recuperado de <https://library.abb.com/es>
- ABB Library. (2012). *Application note. AC500 PLC AND ABB AS355 Drive vía Modbus RTU with ABB standard Library*. Recuperado de <https://library.abb.com/en>
- ABB Library. (2013). *Quickstart Guide. ABB PLC and drives integration using Modbus RTU*. Recuperado de <https://library.abb.com/en>
- ABB Library. (2019). *AC500 PLC. System Assembly and Device Specifications for AC500 v2 Products*. Recuperado de <https://library.abb.com/en>
- ABB Library. (2018). *ACS480 Programa de control estándar. Manual de Firmware*. Recuperado de <https://library.abb.com/es>
- ABB Library. (2018). *Convertidores de frecuencia ACS480. Manual de hardware*. Recuperado de <https://library.abb.com/es>
- ABB Library. (2018). *Convertidores de propósito general. ACS480, de 0.75 a 22kW*. Recuperado de <https://library.abb.com/es>
- ABB Library. (2016). *Tecnología Industrial ABB. Aplicaciones Motion*. Recuperado de <https://library.abb.com/es>
- ABB Library. (2019). *Data Sheet. PM573. Processor Module*. Recuperado de <https://library.abb.com/en>
- ABB Library. (2019). *Data Sheet. TB511. Terminal Base*. Recuperado de <https://library.abb.com/en>
- ABB Library. (2019). *Data Sheet. DA501. Digital/Analog Input/Output Module*. Recuperado de <https://library.abb.com/en>
- ABB Library. (2016). *Application note. Using the CD522 module for master encoder input*. Recuperado de <https://library.abb.com/en>
- ABB Library. (2019). *Data Sheet. CD522. Encoder, Counter and PWM module*. Recuperado de <https://library.abb.com/en>
- ABB Library. (2017). *Programming Software for CP600 Control Panels*. Recuperado de <https://library.abb.com/en>
- ABB Library. (2015). *New Control Panels CP600. Communication with AC500*. Recuperado de <https://library.abb.com/en>
- ABB Library. (2018). *Data Sheet. Control Panel CP600. CP630, CP630-WEB*. Recuperado de <https://library.abb.com/en>

- Grupo SUPPRESS. (2016). *Introducción al software de programación Codesys*. Universidad de León.
- Rosendo Caler Rubio (2015). *Análisis y estudio de comunicaciones industriales para implementar arquitectura de comunicaciones estándar en Planta Estándar de Ciclo Combinado*. Universidad de Alcalá.
- Felipe Mateos (2004). *Autómatas programables Visión General*. Universidad de Oviedo.
- Departamento de Ingeniería Eléctrica, Electrónica y de Control. *Controladores Industriales Inteligentes*. Universidad Nacional de Educación a Distancia.
- José Luis Medina, David Romero, Juan Antonio Gallardo. *Introducción a los autómatas programables*. Universitat Politècnica de Catalunya.
- Ramón Sarraté. *Comunicacions Industrials*. Universitat Politècnica de Catalunya.
- Marc Alfocea. (2019). *Control y Supervisión de un variador de velocidad con un autómata y un sistema HMI*. Universitat Politècnica de Catalunya.
- Ned Mohan (2009). *Electrónica de Potencia, convertidores aplicaciones y diseño*. McGrawHill

15. ANEXOS

15.1. Introducción a las herramientas software

El objetivo de este apartado es tratar brevemente la descarga e instalación, de cada uno de los programas empelados para el desarrollo del proyecto y realizar una introducción sobre cada uno de los entornos.

15.1.1. Automation Builder

15.1.1.1. Descarga e instalación

Este software está disponible en la página web de ABB (<https://new.abb.com/plc/automationbuilder/platform/software>) en tres versiones diferentes. Para la realización de este proyecto se ha empelado la versión básica, que es la gratuita. Los requisitos mínimos del sistema para poder ejecutar le programa son los siguientes:

- Pentium PC, 1 GHz, 3 GB RAM
- Memoria de disco duro de 10 GB.
- Tarjeta gráfica SVGA 256 colores y resolución 1024x768 píxel.
- Puerto USB 2.0.
- Windows XP, Windows 7 (32/64 bits) o superior.

La versión con la que se ha realizado el presente proyecto es la 2.2.4.631, pero siempre es recomendable descargar la última versión disponible.

15.1.1.2. Descripción del entorno

El entorno de Automation Builder se puede dividir en 5 partes diferenciadas a través de las cuales se puede desarrollar cualquier proyecto. En la figura que se muestra a continuación se pueden apreciar las diferentes partes.

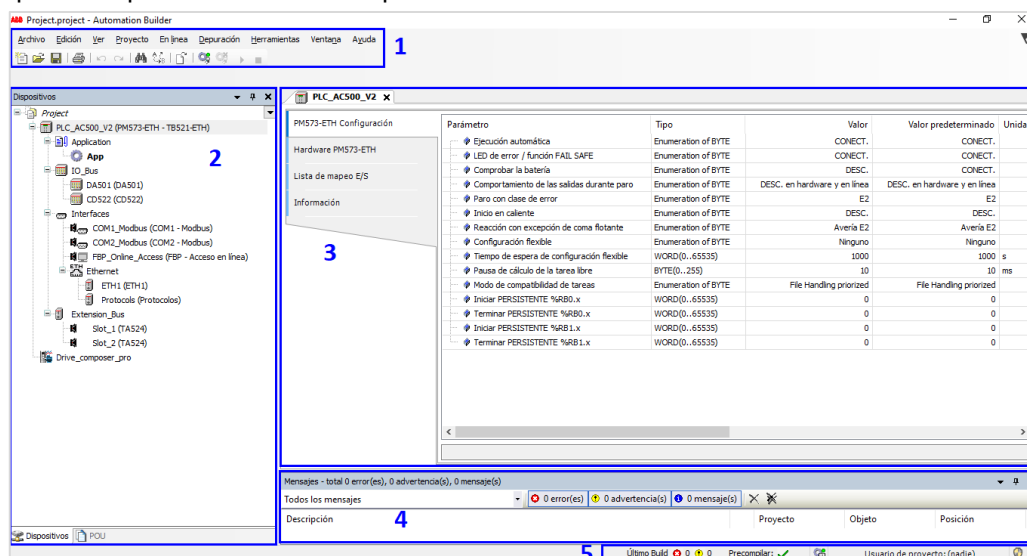


Figura 56: Entorno de Automation Builder

En el cuadro número 1 se puede apreciar la barra de menú y los iconos que permiten acceder a las funciones más comunes rápidamente. Esta parte permite la gestión del programa y acceder a diversas funcionalidades del software. A continuación se detallan algunas funcionalidades más importantes de los diferentes menús:

- Archivo: Es el menú que permite la gestión del archivo de proyecto. Permite crear un nuevo proyecto, cerrar, guardar, imprimir y gestionar configuraciones de página.
- Edición: Permite opciones de edición del proyecto como deshacer, rehacer y buscar y reemplazar.
- Ver: Ofrece el acceso directo a diversas partes del software.
- Proyecto: Permite la configuración del proyecto y realizar tareas relacionadas con el conjunto de dispositivos. También se encuentran las opciones de Importar/Exportar que permiten generar archivos en diferentes formatos.
- En línea: Este menú gestiona la comunicación con el autómatas. Permite conectarse al autómatas iniciando sesión en él, cargar el proyecto en el autómatas o hacer un reset en el mismo.
- Depuración: Permite el inicio del programa en el autómatas. Para poder realizar esta acción es necesario estar conectado al autómatas previamente.
- Herramientas: Permite el acceso a las diferentes herramientas del software. Entre ellas destacar Installation manager que permite ampliar los paquetes de instalación o actualizar los equipos disponibles para añadir al proyecto.
- Ayuda: Permite el acceso a un documento que contiene información relacionada con Automation Builder, Codesys, dispositivos de ABB, herramientas y ejemplos de programación. Es de gran ayuda y utilidad para los usuarios noveles con los programas de ABB.

En el cuadro número 2 se muestra en árbol de dispositivos del proyecto. En él se encuentran todos los dispositivos, comunicaciones y elementos que se hayan añadido al proyecto y permite acceder a las características de cada uno de ellos. Este menú también permite acceder al programa del autómatas mediante la selección del icono Application. En la parte inferior del menú se encuentra la pestaña POU a través de la cual se accede a otro menú en forma de árbol que permite visualizar información del proyecto y acceder a su configuración.

El cuadro 3 es la parte principal del programa. Esta se organiza en pestañas que abren las ventanas de los elementos seleccionados del árbol de dispositivos. En algunos casos ventanas disponen de submenús que permiten el acceso a información y/o funciones adicionales.

En el cuadro 4 proporciona información del estado de los procesos de configuración, guardado y compilación. En caso de que algún proceso no sea correcto, se cataloga como error, advertencia o mensaje en función de su transcendencia.

En el cuadro 5 se muestra un breve resumen con información del estado del proyecto y de la comunicación con el autómata. Se puede observar cuando se ha producido la última compilación del programa, el número de errores y advertencias y si se ha iniciado sesión en el autómata.

15.1.1.3. Creación de un proyecto

Para crear un nuevo proyecto hay que iniciar el software Automation Builder y una vez se haya iniciado, en la pantalla principal se debe seleccionar la opción *Nuevo proyecto*. Para crearlo desde un proyecto hay que dirigirse a la barra menú, clicar *Archivo* y posteriormente seleccionar *Nuevo proyecto*.

A continuación se abrirá una nueva ventana donde se puede seleccionar una plantilla en el caso que se utiliza un autómata de la gama AC500 (1) o un proyecto vacío en caso contrario. Seguidamente se pueden asignar el *Nombre* y la *Ubicación* del proyecto (2) deseados. Para continuar seleccionar *Aceptar* (3).

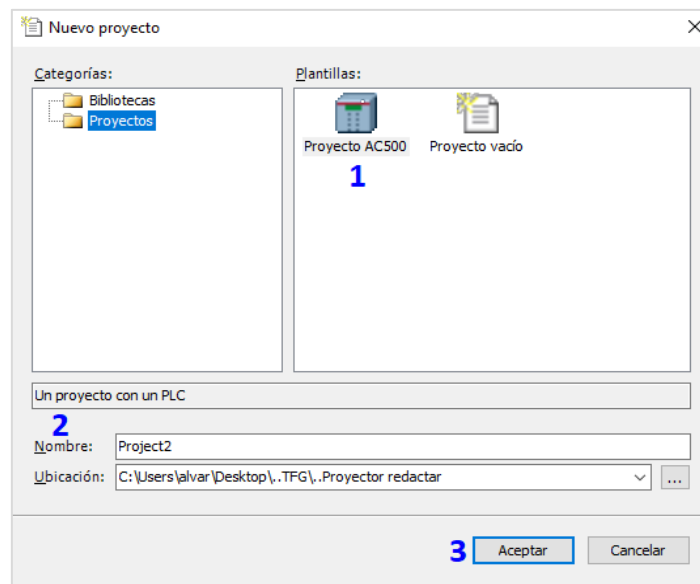


Figura 57: Crear un proyecto nuevo Automation Builder

Se abrirá una nueva ventana en la que se selecciona el autómata del proyecto. En primer lugar hay que seleccionar la familia del autómata. Hecho esto aparecerán los modelos disponibles. Para este proyecto se selecciona la familia de *PLC AC500* (4) y el modelo *PM573-ETH* (5). A continuación es posible modificar el nombre con el que se visualizará el objeto en el proyecto (6). Finalmente para añadirlo hay que hacer clic en *Agregar PLC* (7).

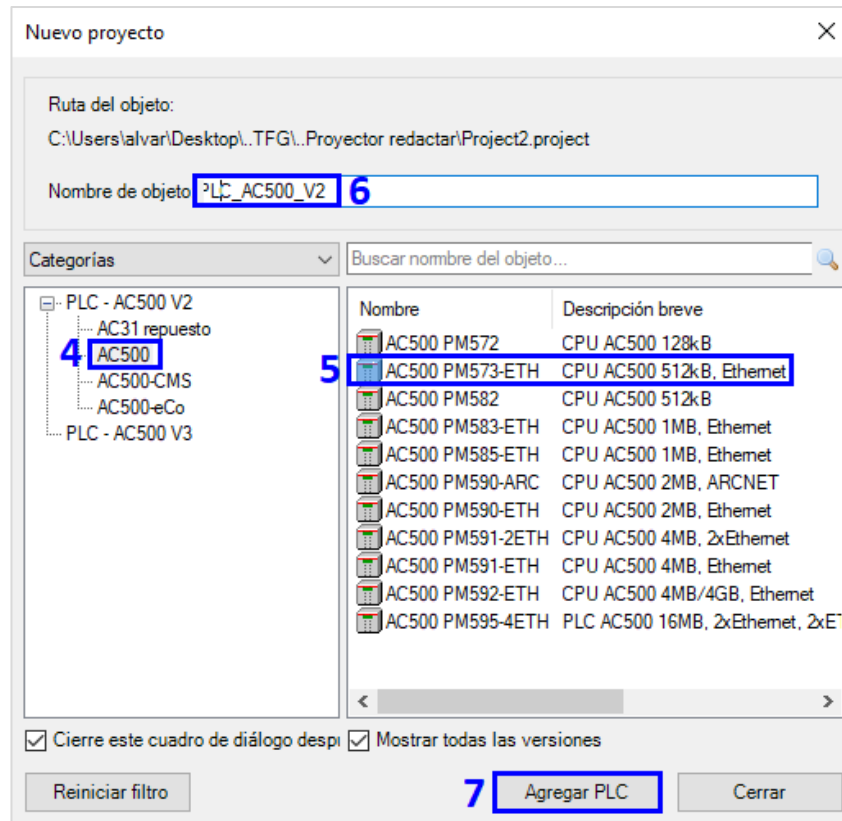


Figura 58: Selección automática

Con este proceso se finaliza la fase inicial para la creación de un nuevo proyecto y por lo tanto se abrirá la ventana principal del software. Se puede observar que en el menú en forma de árbol aparece el autómata seleccionado en el proceso anterior.

A continuación, es recomendable añadir el resto de componentes que forman el proyecto por tal de completar el sistema. Para añadir los módulos de expansión del autómata es preciso hacer clic con el botón derecho en *I/O BUS* y seleccionar *Agregar objeto*. A continuación se abre una nueva ventana en la que se encuentran todos los módulos de extensión disponibles para autómatas de ABB. Para encontrar el módulo requerido se puede restringir la búsqueda mediante el menú en forma de árbol situado a la izquierda (8), buscar por el nombre del módulo en la parte superior (9) o navegar por la lista completa de dispositivos (10). Una vez encontrado el módulo, hay que seleccionarlo y hacer clic en *Agregar objeto* (11). Habrá que repetir este proceso para cada módulo que se desee agregar al autómata. En el presente proyecto se han agregado los módulos DA501 y CD522.

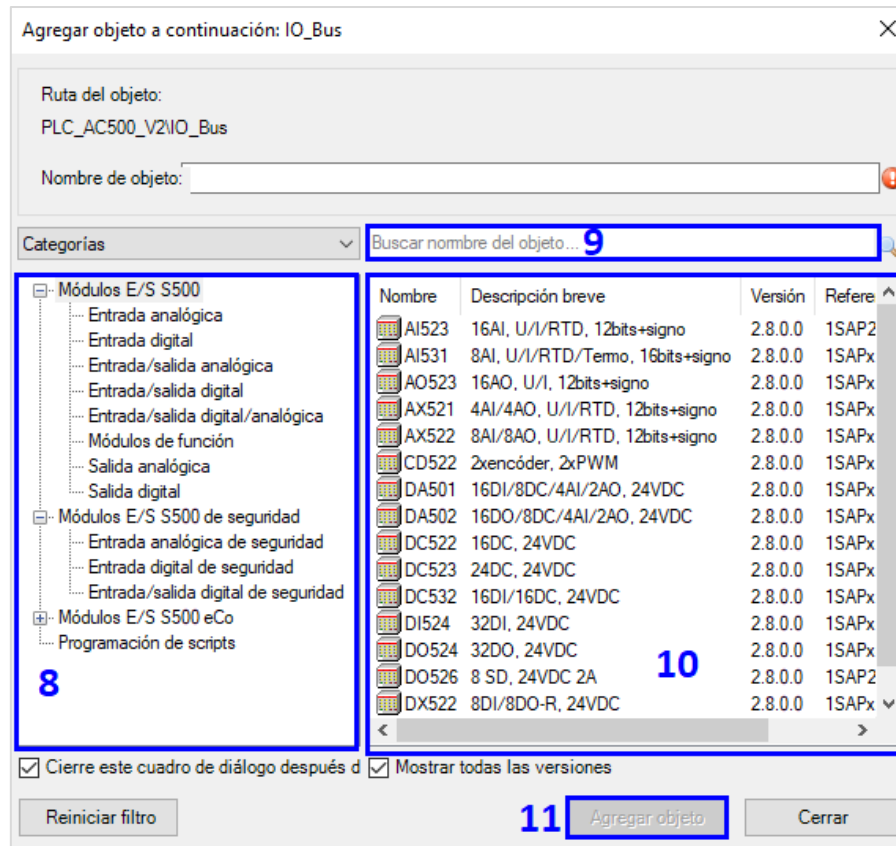


Figura 59: Agregar módulos de extensión al autómeta

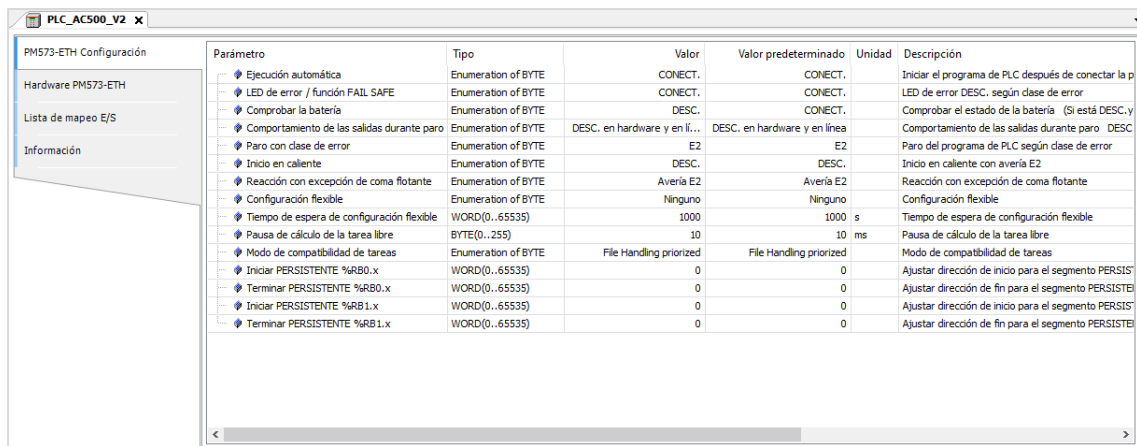
Este proceso es similar para añadir cualquier otro objeto que se desee al proyecto como otro autómeta o un panel de control. La diferencia radica en el elemento que se selecciona para añadir el objeto.

15.1.1.4. Configuración del autómeta

Automation Builder ofrece la posibilidad de configurar parámetros que permiten adaptar cada módulo del autómeta a la funcionalidad deseada. Este apartado trata de forma genérica la configuración de todos los módulos debido a la similitud en el proceso y la gran variedad de módulos de los que se dispone, pero las imágenes corresponden al módulo de la CPU porque es el elemento común en todos los proyectos.

Para acceder a la configuración de un módulo hay que hacer doble clic sobre él en el menú árbol y se abrirá una ventana en la pantalla principal del software. La ventana dispone de cuatro submenús diferentes, configuración, hardware, lista de mapeo de las entradas y salidas e información.

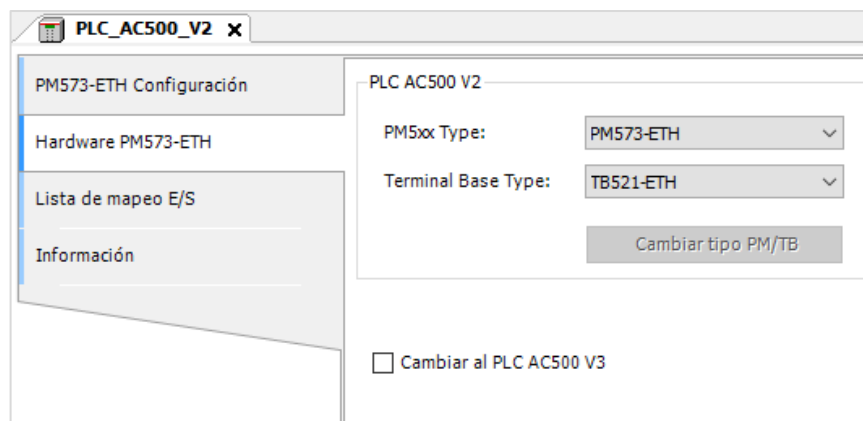
En el menú de configuración se dispone de una lista con diferentes parámetros modificables del módulo seleccionado. Se muestra información relativa de cada parámetro como el tipo de dato, su valor actual y su valor predeterminado, las unidades y su descripción.



Parámetro	Tipo	Valor	Valor predeterminado	Unidad	Descripción
Ejecución automática	Enumeration of BYTE	CONNECT.	CONNECT.		Iniciar el programa de PLC después de conectar la p...
LED de error / función FAIL SAFE	Enumeration of BYTE	CONNECT.	CONNECT.		LED de error DESC., según clase de error
Comprobar la batería	Enumeration of BYTE	DESC.	CONNECT.		Comprobar el estado de la batería (Si está DESC. y
Comportamiento de las salidas durante paro	Enumeration of BYTE	DESC. en hardware y en li...	DESC. en hardware y en línea		Comportamiento de las salidas durante paro DESC
Paro con clase de error	Enumeration of BYTE	E2	E2		Paro del programa de PLC según clase de error
Inicio en caliente	Enumeration of BYTE	DESC.	DESC.		Inicio en caliente con avería E2
Reacción con excepción de coma flotante	Enumeration of BYTE	Avería E2	Avería E2		Reacción con excepción de coma flotante
Configuración flexible	Enumeration of BYTE	Ninguno	Ninguno		Configuración flexible
Tiempo de espera de configuración flexible	WORD(0...65535)	1000	1000	s	Tiempo de espera de configuración flexible
Pausa de cálculo de la tarea libre	BYTE(0...255)	10	10	ms	Pausa de cálculo de la tarea libre
Modo de compatibilidad de tareas	Enumeration of BYTE	File Handling prioritized	File Handling prioritized		Modo de compatibilidad de tareas
Iniciar PERSISTENTE %RB0.x	WORD(0...65535)	0	0		Ajustar dirección de inicio para el segmento PERSIS
Terminar PERSISTENTE %RB0.x	WORD(0...65535)	0	0		Ajustar dirección de fin para el segmento PERSISTE
Iniciar PERSISTENTE %RB1.x	WORD(0...65535)	0	0		Ajustar dirección de inicio para el segmento PERSIS
Terminar PERSISTENTE %RB1.x	WORD(0...65535)	0	0		Ajustar dirección de fin para el segmento PERSISTE

Figura 60: Configuración PM573-ETH

El submenú de hardware permite modificar la CPU seleccionada y su terminal base.



PLC AC500 V2

PM5xx Type: PM573-ETH

Terminal Base Type: TB521-ETH

Cambiar tipo PM/TB

☐ Cambiar al PLC AC500 V3

Figura 61: Hardware PM573-ETH

El submenú *Lista de mapeo de E/S*, muestra información relativa a las entradas y/o salidas del módulo en cuestión como su dirección, el tipo y el terminal al que corresponden. También permite asignarles variables.

En la pestaña de información se muestra información de carácter general del módulo y permite acceder a información más técnica mediante enlaces a la página web de ABB.

15.1.1.5. Bibliotecas

Una de las principales funciones de Automation Builder es integrar todos los dispositivos de un proyecto de automatización en un mismo software. Para ello, dispone de bibliotecas adicionales que generan nuevos recursos de programación para dispositivos como convertidores, robots o sistemas de control predefinidos. Generalmente tienen el fin de facilitar la comunicación entre dispositivos simplificando su programación. Sin embargo, algunas de estas bibliotecas únicamente están destinadas a productos de ABB.

Automation Builder dispone de un instalador de paquetes adicionales a través del cual se pueden añadir librerías al proyecto. Este instalador se encuentra en la opción de *Installation*

Manager dentro del menú *Herramientas*. No obstante, algunas librerías específicas no se encuentran en este instalador y deben descargarse directamente de la página web oficial de ABB.

En este proyecto se ha utilizado la librería *PS553-DRIVES* para establecer las comunicaciones mediante Modbus RTU entre el convertidor de frecuencia y el autómatas. Esta librería contiene bloques de funciones que permiten la comunicación directa, el control básico y acceso a parámetros.

Para descargarse la librería mencionada es necesario acceder a la página web de ABB. El archivo que la contiene es de tipo .zip y para iniciar su instalación se debe ejecutar el archivo *setup.exe*. En este proceso es importante realizar la instalación en la carpeta de la librería, por que posteriormente Automation Builder realizará la búsqueda en la ubicación comentada.

Una vez finalizada la descarga e instalación, se debe cargar la biblioteca en el software Codesys. Para ello, hay que iniciar Codesys y dirigirse a la pestaña *Recursos* (1). A continuación hay que seleccionar la opción *Administrador de bibliotecas* (2) y se mostrarán las librerías cargadas. Posteriormente hay que hacer clic con el botón derecho del ratón en la librería *SysExt_AC500_V10.lib 22.7.19* y seleccionar *Otras bibliotecas...*, momento el cual se abrirá una ventana en la cual habrá que buscar la librería descargada con anterioridad.

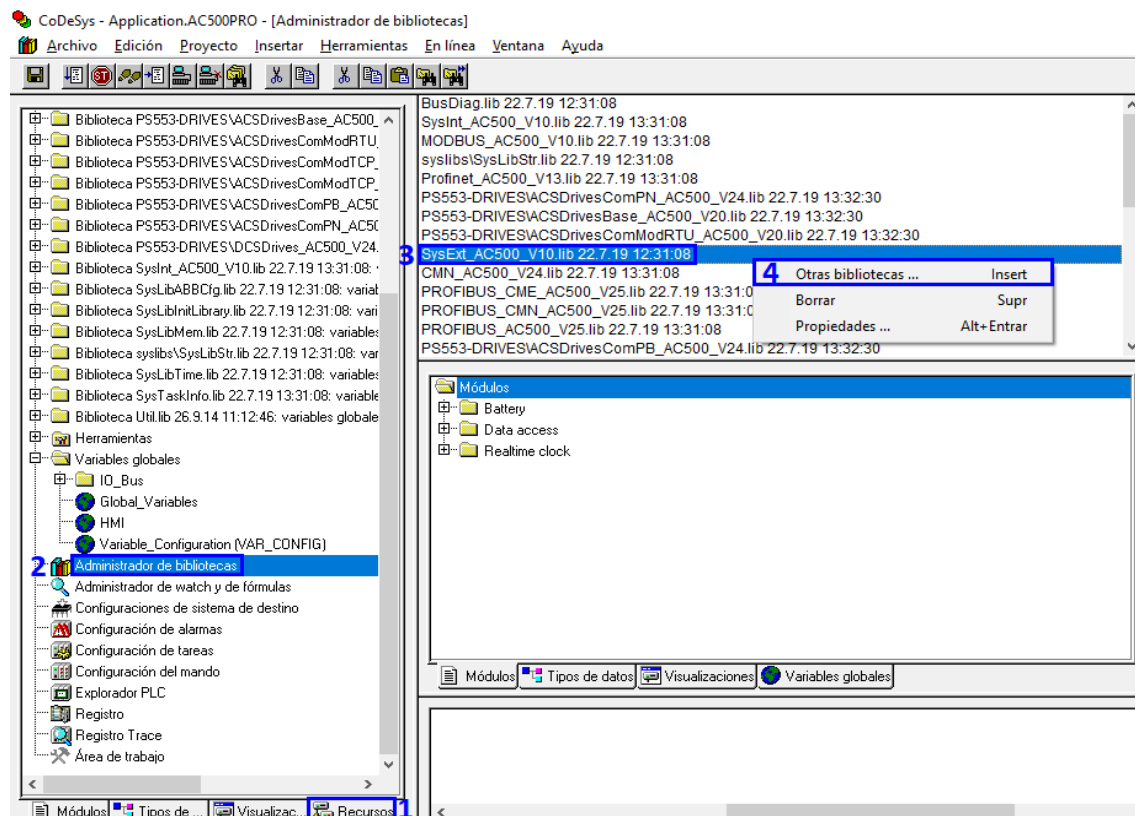


Figura 62: Adición bibliotecas Codesys

Una vez añadida la biblioteca esta se mostrará en el árbol de bibliotecas del proyecto. A partir de este momento las nuevas funcionalidades ya podrán ser utilizadas en la programación del editor.

15.1.2. Codesys

15.1.2.1. Descarga e instalación

Este software libre está incorporado en Automation Builder por lo que no es necesaria su descarga ni instalación. Para iniciar Codesys desde Automation Builder hay que hacer doble clic sobre la pestaña de *Application* en el menú en forma de árbol.

15.1.2.2. Descripción del entorno

El entorno de Codesys se puede dividir en cuatro partes diferenciadas a través de las cuales se pueden realizar los proyectos. A continuación se muestra una figura donde se pueden observar las diferentes partes.

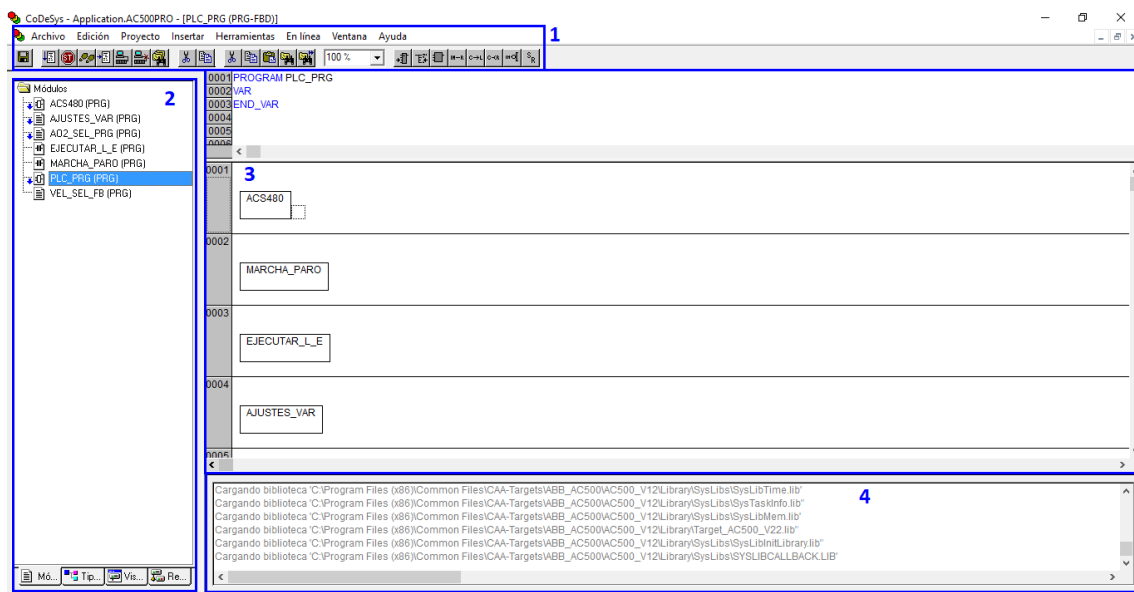


Figura 63: Entorno de Codesys

En el cuadro número 1 se dispone de la barra menú y los iconos de inicio rápido. Esta parte del programa se encarga de la gestión del proyecto y desde ella se puede acceder a las funcionalidades del software. A continuación se describen las funcionalidades de los menús.

- **Archivo:** Permite la gestión del archivo. Únicamente dispone de las funciones de guardar e imprimir. El resto de funcionalidades típicas de este menú se realizan desde Automation Builder.
- **Edición:** Dispone de las herramientas para la edición del programa, como deshacer, rehacer, copiar, pegar o la navegación por los errores de programa.
- **Proyecto:** Permite la gestión del proyecto de programación. En este menú se realizan funciones como la compilación, exportar/importar el programa en otras extensiones y acceder a las opciones del proyecto.
- **Insertar:** Dispone de distintos elementos para insertar el proyecto, en su mayoría elementos gráficos. Estas funciones también se pueden realizar desde los iconos del menú de inicio rápido.

- Herramientas: Proporciona elementos para los proyectos de programación y permite su configuración.
- En línea: Dispone de las opciones de conexión con el autómatas y permite acceder al modo simulación.
- Ventana: Permite modificar la disposición de las ventanas y de las herramientas de visualización.
- Ayuda: Dispone un documento de ayuda al usuario proporciona información de la versión del software.

En el cuadro número 2 se dispone del menú en forma de árbol con diferentes pestañas en su parte inferior. En la pestaña *Módulos* se muestran los diferentes programas del proyecto. En la pestaña *Tipos de datos* se pueden crear tipos de variables. La pestaña *Visualizaciones* se muestran los diferentes programas visuales relacionados con los módulos del proyecto. Por último, en la pestaña *Recursos* se dispone de las opciones del autómatas y del proyecto para realizar distintas modificaciones. De este apartado destacar la pestaña *Variables Globales* que permite crear las variables globales para todo el proyecto.

En el cuadro 3 se encuentra el editor de programación el cual se divide en dos partes diferenciadas. En la parte superior se pueden declarar variables de programa locales para este programa y/o función. La parte inferior permite la creación del código de programa.

En el cuadro número 4 se observan las comunicaciones del software. Se muestra el estado de procesos como el compilado, las acciones realizadas o los errores de programación.

15.1.2.3. Declaración de variables

La declaración de variables en este software puede ser global o local. Las variables locales se definen en la parte superior de cada programa y únicamente se podrá acceder a ellas desde el módulo en el que han sido creadas. Sin embargo, las variables globales se definen en la pestaña *Variables Globales* dentro de la pestaña *Recursos* del menú en forma de árbol y se puede acceder a ellas desde cualquier módulo del proyecto. Es posible generar nuevos archivos de variables globales haciendo clic derecho en *Variables Globales* y seleccionando *Insertar objeto*.

Para la utilización de las variables definidas en Codesys en otro software de programación, se permite la exportación de las variables globales en diferentes extensiones, pero estas deben ser guardadas en la memoria del autómatas.

La declaración de las variables locales y globales se realiza sintácticamente del mismo modo. Primeramente se debe introducir el nombre de la variable, seguidamente dos puntos y a continuación el tipo de variable (siempre en mayúsculas). Hay que finalizar la sentencia con punto y coma. Es posible otorgar un valor inicial a la variable, para ello, antes del punto y comas se debe escribir dos puntos e igual seguido del valor.

Si se pretende exportar las variables globales para ser utilizadas por un bus de campo, estas deben ser asignadas a una memoria interna. Para ello después del nombre de la variable se debe introducir la palabra "AT" seguida del tipo de memoria y dirección.

Existen diferentes memorias para cada tipo de variable y depende de la información que desea almacenar. A continuación se muestra la declaración de una variable de tipo booleano (memoria de un bit) y una de tipo entero (memoria de 8 bits) con la dirección de memoria incluida.

- Tipo BOOL: MARCHA AT %MX0.0.1:BOOL:=FALSE;
- Tipo INT: VELOCIDAD AT %MW0.1:INT:=1000;

Codesys dispone de un método más dinámico para definir las variables a medida que se realiza un programa. Durante la realización del código, si este detecta que alguna palabra no corresponde a una variable anteriormente declarada ni a ninguna sentencia, aparece un cuadro de diálogo que permite declarar la variable y seleccionar todos los parámetros correspondientes. De este modo la se declara la variable automáticamente en la lista de variables seleccionada.

Figura 64: Declaración variables Codesys

15.1.2.4. Exportar variables

Para exportar una lista de variable globales únicamente hay que hacer clic derecho sobre la lista en cuestión y seleccionar la opción *Exportar objeto....* Posteriormente se abrirá una ventana que permitirá guardar el archivo en la ubicación deseada y con la extensión deseada. Es importante que la extensión del archivo seleccionada se pueda interpretar desde el software que posteriormente realizará la importación del archivo. En el presente proyecto se ha seleccionado la extensión *.exp* que es interpretable por el panel de control.

15.1.2.5. Lenguajes de programación

Codesys permite la programación en seis lenguajes diferentes. Cada módulo del programa es independiente de los demás y únicamente permite el uso de un lenguaje. Pero los diferentes módulos interaccionan entre ellos, lo que permite el desarrollo de un programa general en diferentes lenguajes.

Con esta variedad de lenguajes, es importante seleccionar correctamente el lenguaje que se usará para llevar a cabo cada tarea, puesto que cada lenguaje tiene sus particularidades que facilitan o dificultan su programación.

Los lenguajes mediante los cuales Codesys permite la programación son los siguientes:

- Lista de instrucciones (IL): Se compone de una serie de expresiones lógicas y booleanas escritas de manera secuencial. Tiene estructura en forma de lista, donde cada fila es una instrucción independiente. La estructura de cada línea consta de número de instrucción, operador y operando. Este lenguaje es útil si se busca que las instrucciones se ejecuten lo más rápido posible.
- Ladder (LD): Este lenguaje se basa en la lógica de contactos donde estos únicamente pueden tener dos estado, abierto o cerrado. Se compone por barras de alimentación donde se representan las variables lógicas mediante el uso de bobinas y relés. También es posible la utilización de temporizadores, contadores y añadir bloques funcionales complejos.
- Diagrama de bloques funcionales (FBD): Es un lenguaje simbólico representado por estructuras en forma de bloques. Estos bloques representan un operando lógico o una función. La variable de entrada se combina con los bloques hasta que llega a la salida.
- Bloques de funciones secuenciales (SFC): Se basa en el lenguaje Grafcet. Cada acción dispone de una condición previa para su activación, de modo que las acciones se asocian a pasos o transiciones. Permite una visión completa del programa, pero al mismo tiempo descompone cada acción en pequeñas partes. Cada bloque funcional se puede programar en un lenguaje diferente.
- Texto estructurado (ST): Es un lenguaje de alto nivel, basado Pascal y C. Es un lenguaje estructurado en bloques y se utiliza para el desarrollo de expresiones complejas. Permite la ejecución condicional, en bucles y funciones.
- Continuous Function Chart (CFC): Este lenguaje no está incluido en el estándar IEC61131-3. Es una variación del lenguaje de bloques funcionales (FBD). Los bloques son predefinidos para la realización de aplicaciones secuenciales y repetitivas. Este lenguaje no es compatible con los demás, y por lo tanto, no es posible su traducción automática.
-

15.1.3. Panel Builder 600

15.1.3.1. Descarga e instalación

Este software está integrado en Automation Builder, por lo tanto, si se ha descargado este software no es necesario realizar su descarga e instalación. Únicamente es posible incorporar el proyecto de Panel Builder 600 en Automation Builder en el caso que se tenga la versión Premium de Automation Builder, por lo tanto, para iniciar Panel Builder 600 es necesario hacerlo desde su propio archivo ejecutable.

15.1.3.2. Descripción del entorno

El software Panel Builder 600 mantiene un entorno parecido al de Automation Builder. Este entorno se puede dividir en cuatro partes diferenciadas en las cuales se disponen todas las funcionalidades del software. A continuación se muestran una figura donde se pueden apreciar las diferentes partes.

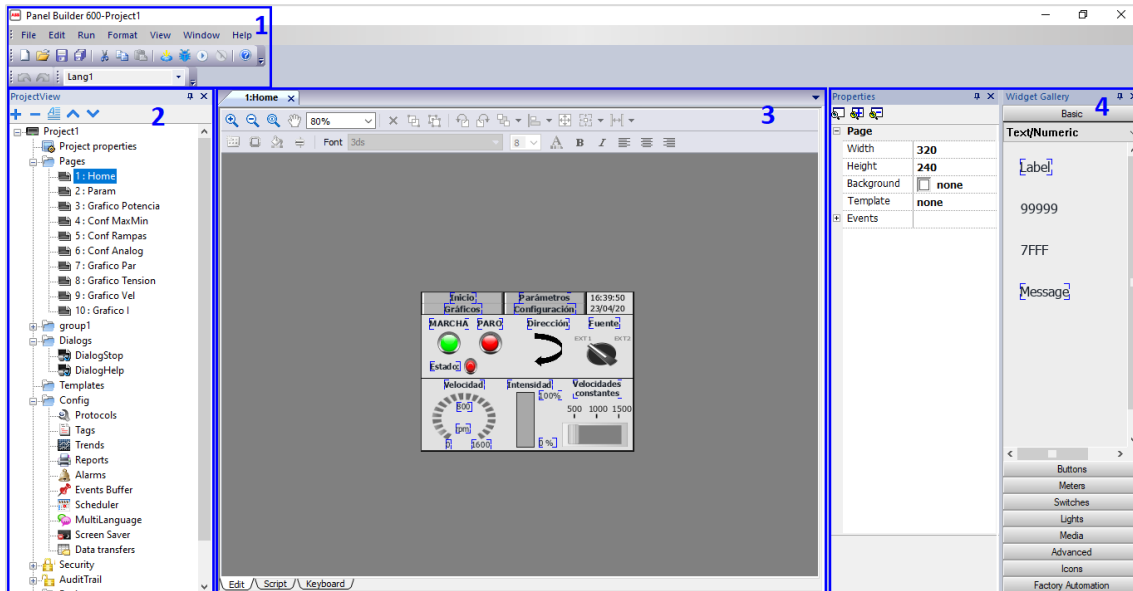


Figura 65: Entorno Automation Builder

En el cuadro número 1 se dispone de la barra menú y de los iconos de inicio rápido. Esta parte permite la gestión del programa y acceder a las funcionalidades del software. A continuación se muestran las funcionalidades de los menús:

- File: Permite la gestión de archivos. Entre sus funciones destacan las de crear nuevos proyectos o abrir proyectos ya existentes o guardar el proyecto actual.
- Edit: Dispone de las herramientas de edición como deshacer, rehacer, cortar y copiar.
- Run: Este menú permite la conexión y gestión del HMI. También permite acceder al modo de simulación.
- View: Proporciona la posibilidad de gestionar la apariencia del entorno y de personalización de la barra de herramientas.
- Window: Gestiona las diversas ventanas del proyecto.
- Help: Dispone de un documento de ayuda que proporciona información sobre como programar el panel y la compatibilidad del software con diferentes dispositivos.

En el cuadro 2 se encuentra el árbol del proyecto. En este apartado se muestran las páginas y cuadros de diálogo del proyecto, y permite acceder a las herramientas de comunicación y funcionalidades del panel de control.

En el cuadro número 3 se muestra la pantalla según el tamaño configurado y es el área donde se realiza el diseño y configuración. En la parte superior dispone de unos elementos básicos de visualización y edición.

En el cuadro 4 se dispone de dos pestañas. En la *Widget Gallery* se encuentran todos los elementos gráficos que permiten el diseño de la pantalla. En *Properties* se encuentran todas las propiedades modificables del widget seleccionado.

15.1.3.3. Creación de un proyecto

Para crear un proyecto, en primer lugar se debe iniciar el software Panel Builder 600. Una vez se ha abierto el programa, hay que seleccionar el icono de la hoja en blanco *New*. A continuación se abrirá una ventana donde se debe introducir el nombre del proyecto (1) y su ubicación (2). Para proseguir con la configuración de proyecto seleccionar *Next* (3). Seguidamente se abrirá una nueva ventana donde hay que seleccionar la pantalla que se usará para la realización del proyecto (4), de este modo el software establece las dimensiones de la pantalla automáticamente. Después hay que seleccionar la orientación del panel (5) y para finalizar hacer clic en *Finish* (6).

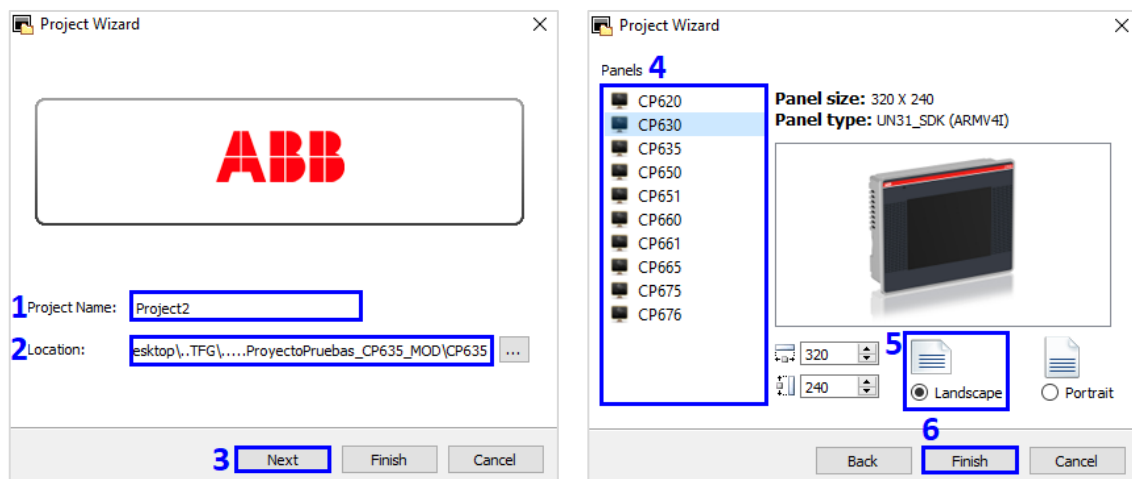


Figura 66: Crear un proyecto nuevo Panel Builder 600

15.1.3.4. Importar variables

El software Panel Builder 600 permite la creación e importación de variables para su uso y ejecución. Todas las variables del HMI utilizadas en el presente proyecto son importadas del autómatas, dado que su función es compartir información entre ambos equipos y se han creado con anterioridad en Automation Builder.

Para poder importar variables primero es necesario haberlas exportado desde el software del autómatas, en este caso Codesys. Es necesario que las variables indiquen el tipo y la posición e memoria donde se guarden. Esto permite que a través del bus de campo, la lectura o escritura de las variables se ejecute sobre una dirección.

Antes de realizar la importación de variables es necesario haber configurado un protocolo de comunicaciones. Hecho esto, se puede proceder a la importación de variables. Para ello hay que seleccionar la pestaña *Tags* (1) y posteriormente hacer clic el icono de *Import Tags* (2). A continuación se abrirá una ventana de diálogo en la que hay que seleccionar el protocolo (4) y el tipo de archivo (4) a importar. Para continuar hay que presionar *OK* (5) y se abrirá una nueva ventana en la que hay que buscar la ubicación de archivo y abrirlo.

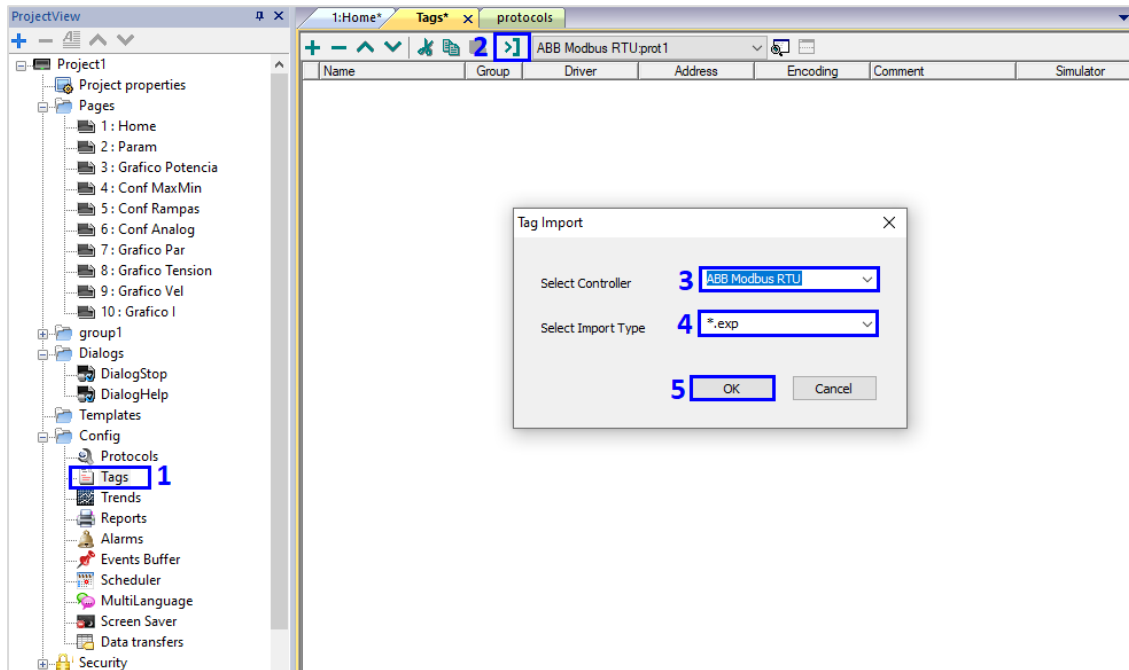


Figura 67: Importación variables Panel Builder

Una vez se haya abierto el archivo, se volverá a la página inicial. En este momento la ventana anterior se dividirá en dos partes. En la parte inferior se mostrarán las variables declaradas en el archivo importado (6) y en la parte superior las variables declaradas en Panel Builder. Para traspasar las variables al programa actual, hay que seleccionar las que se deseen traspasar y en la barra de tareas inferior presionar *Import Tags* (7). Si el proceso se ha realizado correctamente, estas variables se ubicarán automáticamente en la parte superior (8). En este momento estas variables ya estarán declaradas en Panel Builder y por lo tanto estarán disponibles para su uso en el programa.

Name	Group	Driver	Address	Encoding	Comment	Simulator
MARCHA		ABB Modbus RTU:pi 1	MX0 0.1 0	boolean		Variables
PARO		ABB Modbus RTU:pi 1	MX0 0.2 0	boolean		Variables
DIRECCION		ABB Modbus RTU:pi 1	MX0 0.3 0	boolean		Variables
EJECUTAR		ABB Modbus RTU:pi 1	MX0 0.4 0	boolean		Variables
RESET		ABB Modbus RTU:pi 1	MX0 0.5 0	boolean		Variables
CONF_MOD		ABB Modbus RTU:pi 1	MX0 0.6 0	boolean		Variables
ESTADO_CONVRT		ABB Modbus RTU:pi 1	MX0 0.7 0	boolean		Variables
EJECUTAR_EXT		ABB Modbus RTU:pi 1	MX0 1.0 0	boolean		Variables
VELOCIDAD		ABB Modbus RTU:pi 1	MW0 2 0	short		Variables
VEL_100		ABB Modbus RTU:pi 1	MW0 3 0	short		Variables
PAR		ABB Modbus RTU:pi 1	MW0 4 0	short		Variables
PAR_100		ABB Modbus RTU:pi 1	MW0 5 0	short		Variables
U_OUT		ABB Modbus RTU:pi 1	MW0 6 0	short		Variables

Import Tag(s)	memorytype	arrayindex.su...	index	datatype	array	arraysize
PARO	MX0	0.1	0	boolean	false	0
PARO	MX0	0.2	0	boolean	false	0
DIRECCION	MX0	0.3	0	boolean	false	0
EJECUTAR	MX0	0.4	0	boolean	false	0
RESET	MX0	0.5	0	boolean	false	0
CONF_MOD	MX0	0.6	0	boolean	false	0
ESTADO_CONV	MX0	0.7	0	boolean	false	0
EJECUTAR_EX	MX0	1.0	0	boolean	false	0
VELOCIDAD	MW0	2	0	short	false	0
VEL_100	MW0	3	0	short	false	0
PAR	MW0	4	0	short	false	0
PAR_100	MW0	5	0	short	false	0
U_OUT	MW0	6	0	short	false	0
FREC_OUT	MW0	7	0	short	false	0
INTENSIDAD	MW0	8	0	short	false	0
I_100	MW0	9	0	short	false	0
U_BUS_CC	MW0	10	0	short	false	0

Figura 68: Declaración variables Panel Builder

15.2. Control y supervisión del motor

15.2.1. Programación del convertidor

A continuación se muestra una tabla con los parámetros configurados del convertidor de frecuencia y su descripción.

Parámetro	Ajuste seleccionado	Función/Información
Grupo 10 DI, RO estándar – Configuración de las entradas digitales y salidas de relé		
10.24 RO1 fuente	Listo para marcha	Selecciona la señal del convertidor que se conecta a la salida de relé RO1
10.27 RO2 fuente	Aviso	Selecciona la señal del convertidor que se conecta a la salida de relé RO2

Parámetro	Ajuste seleccionado	Función/Información
10.30 RO3 fuente	Fallo	Selecciona la señal del convertidor que se conecta a la salida de relé RO3
Grupo 12 Entradas analógicas estándar – Se configuran los límites y el escalado de AI1. AI2 no se configura porque su configuración se lleva a cabo desde el EXT2 (HMI)		
12.17 AI1 mín	0 V	Define el valor de emplazamiento mínimo para la entrada analógica AI1
12.18 AI1 máx	10 V	Define el valor de emplazamiento máximo para la entrada analógica AI1
12.19 AI1 Escala en AI1 mín	0 rpm	Define el valor real interno que corresponde al valor mínimo de la AI1
12.20 AI1 Escala en AI1 máx	1600 rpm	Define el valor real interno que corresponde al valor máximo de la AI1
Grupo 13 Salidas analógicas estándar – Configuración de la salidas analógicas estándar		
13.12 AO1 Fuente	Frecuencia de salida	Selecciona una señal para conectarla a la salida analógica AO1
13.16 AO1 tiempo de filtro	0,1 s	Define la constante de tiempo de filtro para la AO1
13.19 AO1 fuente mín	0 mA	Define el valor mínimo para la salida analógica AO1
13.20 AO1 fuente máx	20 mA	Valor máximo de la salida analógica AO1

Parámetro	Ajuste seleccionado	Función/Información
13.22 AO2 fuente	Este parámetro se modifica desde el HMI para poder visualizar diferentes parámetros	Selecciona una señal para conectarla a la salida analógica AO2
13.26 AO1 tiempo de filtro	0,1 s	Define la constante de tiempo de filtro para la AO2
13.19 AO1 fuente mín	0 mA	Define el valor mínimo para la salida analógica AO2
13.20 AO1 fuente máx	20 mA	Valor máximo de la salida analógica AO2
Grupo 19 Modo de operación – Selecciona el modo operativo de las fuentes de lugar de control local y externo y los modos de operación. Se realiza un modo de control de la velocidad y la fuente de control se selecciona mediante el HMI		
19.01 Modo de operación actual	Velocidad	Control de velocidad (en modo control vectorial)
19.11 Ext1/Ext2 selección	BCI MCW bit 11 El bit 11 de la palabra de control se recibe a través de la interfaz de bus de campo.	Selecciona la fuente de selección del lugar de control externo EXT1/EXT2 0 Ext1 (Maqueta) 1 Ext2 (HMI)
Grupo 20 Marcha/Paro/Dirección – Selección de las fuentes de señal marcha/paro/dirección y de la interpretación de las transiciones		
20.01 Ext1 Marcha/Paro/Dir	In1 Marcha, In2 Paro In3 Dir	Selecciona la fuente de las órdenes para el lugar de control externo 1 (EXT1)
20.03 Ext1 In1 fuente	Entrada digital DI1	Selecciona la fuente 1 para el parámetro In1 Marcha

Parámetro	Ajuste seleccionado	Función/Información
20.04 Ext1 In2 fuente	Entrada digital DI2	Selecciona la fuente 2 para el parámetro In2 Paro
20.05 Ext1 In3 fuente	Entrada digital DI3	Selecciona la fuente 3 para el parámetro In3 Dirección
20.06 Ext2 Marcha/Paro/Dir	Bus de campo integrado	Selecciona la fuente de las órdenes para el lugar de control externo 2 (EXT2).
20.07 Ext2 tipo de activación	Nivel	Define si la señal de marcha en EXT2 actúa por flanco o nivel
Grupo 22 Selección referencia de velocidad – Selección de la referencia de velocidad y ajustes del potenciómetro del motor		
22.11 Ext1 velocidad Ref1	AI1 escalada	Selecciona la fuente de referencia 1 de velocidad para EXT1
22.18 Ext2 velocidad Ref1	BCI referencia 1	Selecciona la fuente de referencia 1 de velocidad para EXT2
22.21 Velocidad constante función	Modo velocidad constante, 7 velocidades constantes seleccionables	Determina como se seleccionan las velocidades constantes
22.22 Vel constante Sel1	DI4	Parámetro selección velocidades
22.23 Vel constante Sel2	DI5	Parámetro selección velocidad
22.24 Vel constante Sel3	0 (Siempre desactivado)	Parámetro selección velocidad

Parámetro	Ajuste seleccionado	Función/Información
22.26 Vel constante 1	500 rpm	Define la velocidad constante 1
22.27 Vel constante 2	1000 rpm	Define la velocidad constante 2
22.28 Vel constante 3	1500 rpm	Define la velocidad constante 3
Grupo 23 Rampas Acel/Decel Velocidad – Ajustes de la rampa de la aceleración y deceleración de la referencia de velocidad		
23.11 Selección rampa	DI6	Selecciona la fuente que cambia entre dos series de tiempo de rampa
23.12 Tiempo aceleración 1	20 s	Tiempo máximo necesario para que una velocidad incremente a otra mayor
23.13 Tiempo deceleración 1	20 s	Tiempo máximo necesario para que una velocidad disminuya a otra menor
23.14 Tiempo aceleración 2	10 s	Tiempo máximo necesario para que una velocidad incremente a otra mayor
23.15 Tiempo deceleración 2	10 s	Tiempo máximo necesario para que una velocidad disminuya a otra menor
23.23 Paro emergencia tiempo	3 s	Define el tiempo dentro del cual se detiene el convertidor si se activa el paro de emergencia
22.32 Tiempo de forma 1	0 s	Define la forma de las rampas de aceleración y deceleración

Parámetro	Ajuste seleccionado	Función/Información
Grupo 30 Limites – Limites del funcionamiento del convertidor		
30.11 Velocidad mínima	-1600 rpm	Define el valor mínimo de velocidad
30.12 Velocidad máxima	+1600 rpm	Define el valor máximo de velocidad
30.13 Frecuencia mínima	-50 Hz	Define el valor mínimo de frecuencia
30.14 Frecuencia máxima	+50 Hz	Define el valor máximo de frecuencia
30.17 Intensidad máxima	8.72 A	Define el valor máximo de intensidad
30.18 Limite par selección	Serie de límite de par 1	Selecciona una fuente que cambia entre dos series distintas de límites de par
30.19 Par mínimo 1	-300%	Define un límite de par mínimo
30.20 Par máximo 1	+300%	Define un límite de par máximo
303.30 Control sobretensión	Habilitar	Activa el control de sobretensión en el bus de CC intermedio.
30.31 Control subtensión	Habilitar	Activa el control de subtensión en el bus de cc intermedio.
30.35 Limita intensidad térmica	Habilitar	Habilita/deshabilita la limitación de corriente de salida en temperatura.

Parámetro	Ajuste seleccionado	Función/Información
Grupo 46 Ajustes monitorización/escalado – Ajustes de supervisión de velocidad, filtro de señal actual y ajustes de escalado general		
46.01 Escalado velocidad	30000 rpm	Define el escalado de los parámetros relacionados con la velocidad
46.02 Escalado frecuencia	1000 Hz	Define el escalado de los parámetros de par.
46.03 Escalado par	30000%	Define el escalado de los parámetros de par
46.04 Escalado potencia	4kW	Define el escalado de los parámetros de potencia.
46.05 Escalado intensidad	30000 A	Define el escalado de los parámetros de intensidad.
Grupo 58 Bus de campo integrado – Configuración de la interfaz de bus de campo integrado (BCI)		
58.01 Habilitar protocolo	Modbus RTU	Habilita/deshabilita la interfaz de bus de campo integrada y selecciona el protocolo que se debe usar.
58.03 Nodo	Nodo 1	Define el nodo del convertidor en el enlace de bus de campo. Están permitidos los valores 1 a 247.
58.04 Velocidad transmisión	19,2 Kb/s	Selecciona la velocidad de transferencia del enlace de bus de campo.
58.05 Paridad	Ninguno 1 (8 bits de datos, sin bit de paridad y un bit de paro)	Selecciona el ajuste para la paridad y el bit de paro. Hay

Parámetro	Ajuste seleccionado	Función/Información
		que utilizar el mismo ajuste que en la estación maestra.
58.06 Control comunicación	Habilitado	Funcionamiento normal. Asume los ajustes del BCI cambiados durante el funcionamiento.
58.14 Pérdida comunicación	Fallo	Define la medida que se toma cuando se detecta una pérdida de comunicación.
58.33 Modo direccionamiento	Modo 0	Define el mapeo entre parámetros y registros de retención en el rango de registros de Modbus.
Grupo 99 Datos del Motor – Ajustes de configuración del motor		
99.03 Tipo motor	Motor asíncrono	Selección del tipo de motor
99.04 Modo control motor	Vectorial	Selecciona el modo de control del motor
99.06 Intensidad nominal motor	8,72 A	Define la intensidad nominal del motor
99.07 Tensión nominal motor	400 V	Define la tensión nominal suministrada al motor
99.08 Frecuencia nominal motor	50 Hz	Define la frecuencia nominal del motor
99.09 Velocidad nominal motor	1430 rpm	Define la velocidad nominal del motor
99.10 Potencia nominal motor	4 kW	Define la potencia nominal del motor

Parámetro	Ajuste seleccionado	Función/Información
99.11 Cos φ nominal	0.82	Define el coseno de fi nominal del motor
99.15 Pares de polos motor	2	Número calculado de pares de polos del motor
99.16 Orden fases motor	U V W	Conmuta el sentido de giro del motor

Tabla 18: Configuración parámetros ACS480

15.2.2. Programación PAC

15.2.2.1. Programa ACS480

Este programa permite las comunicaciones entre el convertidor de frecuencia ACS480 y el autómat. Para su realización se ha descargado la librería PS553-DRIVES de la página web oficial de ABB, la cual, es una librería únicamente para productos de ABB que permite la comunicación directa mediante diferentes protocolos, el control básico y acceso a parámetros del variador. En las entradas de estos bloques es necesario introducir las variables o señales para el funcionamiento del bloque, y en las salidas se puede observar el funcionamiento del bloque. A continuación se muestran y describen los diferentes bloques de funciones empleados en este programa.

15.2.2.1.1. Bloque de funciones ACS_COM_MOD_RTU

Este bloque funcional permite realizar un control básico de las comunicaciones mediante el protocolo Modbus RTU con los dispositivos de las gamas ACS y DCS. En este caso permite la comunicación entre el ACS480 y la CP630.

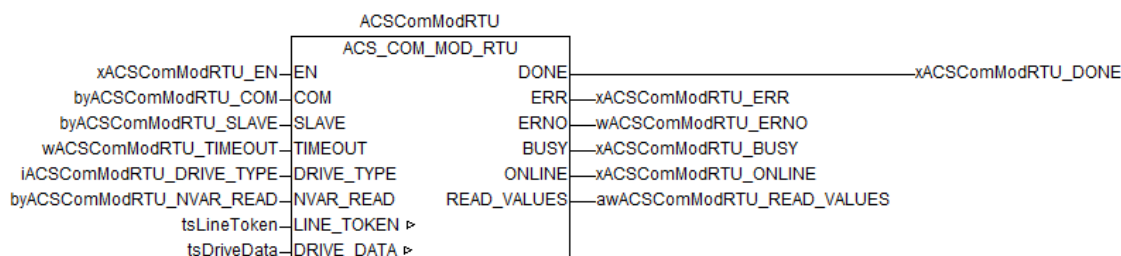


Figura 69: Bloque ACS_COM_MOD_RTU

Parámetro	Ajuste seleccionado	Función/Información
EN	TRUE	Habilita (TRUE) /deshabilita (FALSE) el bloque de funciones Tipo de dato: BOOL
COM	1 : COM1	Interfaz del bus de campo Modbus Tipo de dato: BYTE
SLAVE	Nodo 1	Dirección del nodo del esclavo. Tipo de dato: BYTE
TIMEOUT	1000 ms	Tiempo de espera en recibir una respuesta. Tipo de dato: WORD
DRIVES_TYPE	16 (ACS480)	Tipo de convertidor Tipo de dato: INT
NVAR_READ	0	Numero de variables a leer. Variable no disponible para ACS480. Tipo de dato: BYTE
LINE_TOKEN	MODBUD_TOKEN_COM1 (variable de programa)	Esta variable contiene el token de la línea Modbus Tipo de dato: ACS_MOD_TOKEN_TYPE
DRIVE_DATA	DRIVE_POINTER (variable de programa)	Esta variable contiene la información de la unidad de control Tipo de dato: ACS_DRIVE_DATA_TYPE

Tabla 19: Entradas ACS_COM_MOD_RTU

15.2.2.1.2. Bloque de funciones ACS_DRIVES_CTRL_STANDARD

Este bloque de funciones se usa para controlar convertidores de la gama ACS con dispositivos de la gama ABB. Este proporciona señales de marcha y paro a la unidad de control y diagnósticos de las señales leídas. En las entradas se le proporciona información relacionada con las señales básicas de funcionamiento y referencias, y en la salida se puede observar el estado del control.

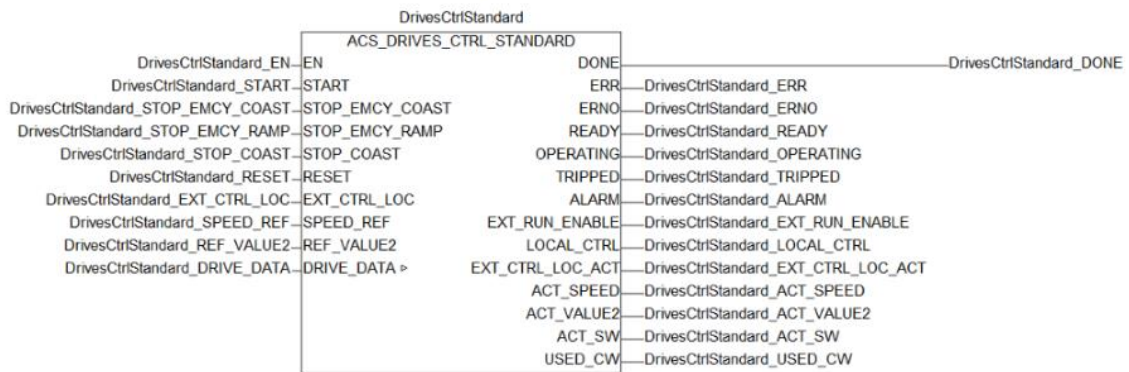


Figura 70: Bloque ACS_DRIVES_CTRL_STANDARD

Parámetro	Ajuste seleccionado	Función/Información
EN	TRUE	Habilita (TRUE) /deshabilita (FALSE) el bloque de funciones Tipo de dato: BOOL
START	ON (variable de programa)	Permite realizar la función de marcha en la unidad de control. Tipo de dato: BOOL
STOP_EMCY_COAST	TRUE	Habilita/deshabilita el paro de emergencia Tipo de dato: BOOL
STOP_EMCY_RAMP	TRUE	Permite realizar la función de paro siguiendo la rampa definida en el convertidor Tipo de dato: BOOL

Parámetro	Ajuste seleccionado	Función/Información
STOP_COAST	PARO (variable de programa)	Permite realizar la función de paro. Tipo de dato: BOOL
RESET	RESET (variable de programa)	Reinicia los avisos y fallos en la unidad de control. Tipo de dato: BOOL
EXT_CTRL_LOC	EXT_SEL_2 (variable de programa)	Selecciona la fuente de control de la unidad de control Tipo de dato: BOOL
SPEED_REF	VEL_REF2 (variable de programa)	Selecciona la velocidad de referencia a proporcionar a la unidad de control Tipo de dato: REAL
REF_VALUE2	-	Proporciona una segunda referencia de velocidad a la unidad de control Tipo de dato: REAL
DRIVE_DATA	DRIVE_POINTER (variable de programa)	Contiene información relacionada con la unidad de control Tipo de dato: ACS_DRIVE_DATA_TYPE

Tabla 20: Entradas ACS_DRIVES_CTRL_STANDARD

15.2.2.1.3. Bloque de funciones ACS_MOD_READ_N_PRM

Este bloque permite la lectura de parámetros del convertidor. Para ello es necesario saber la dirección Modbus en la que se ha codificado el parámetro en cuestión y asociarle una variable de programa en la que se guarde la información.

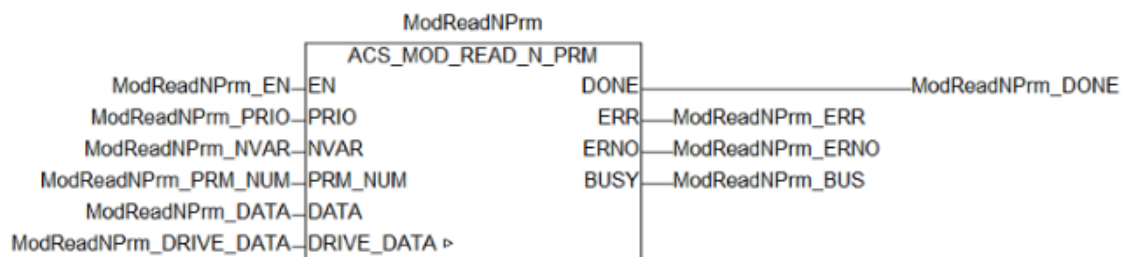


Figura 71: Bloque MOD_READ_N_PRM

Parámetro	Ajuste seleccionado	Función/Información
EN	EJECUTAR_L (variable de programa)	Habilita (TRUE) /deshabilita (FALSE) el bloque de funciones Tipo de dato: BOOL
PRI0	-	Es un parámetro que no tiene uso por el momento. Está reservado para su uso en el futuro. Tipo de dato: BOOL
NVAR	1	Define el número de variables a leer por el bloque. Tipo de dato: UNIT
PRM_NUM	Depende de la variable que se desea leer del convertidor	Cada variable del convertidor está codificada en un parámetro. La fórmula para codificar cada parámetro es la siguiente: Grupo parám. x 100 + n° parám. Por ejemplo, el parámetro 30.20 se codifica como el parámetro 3020. Tipo de dato: UNIT
DATA	La variable de programa en la que interese guardar la información leída	Es la variable en la cual se quiere guardar la información leída. Al tener que estar en formato DWORD, se usa el bloque ADR que es una función de dirección y arroja la dirección de su argumento en DWORD Tipo de dato: DWORD
DRIVE_DATA	DRIVE_POINTER (variable de programa)	Tipo de dato: ACS_DRIVE_DATA_TYPE

Tabla 21: Entradas ACS_MOD_READ_N_PRM

15.2.2.1.4. Bloque de funciones ACS_MOD_WRITE_N_PRM

Este bloque permite la escritura de parámetros en el convertidor. Para ello es necesario saber la dirección Modbus en la que se ha codificado el parámetro en cuestión y e indicar la variable de la cual tiene que copiar el valor.

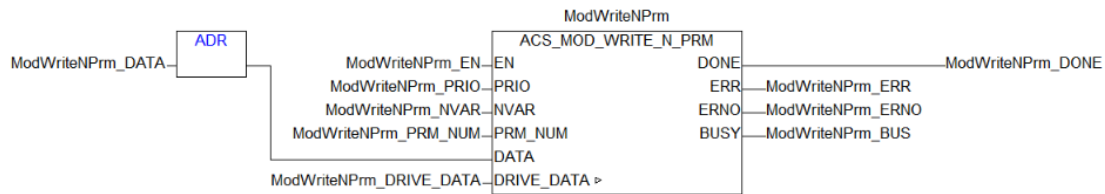


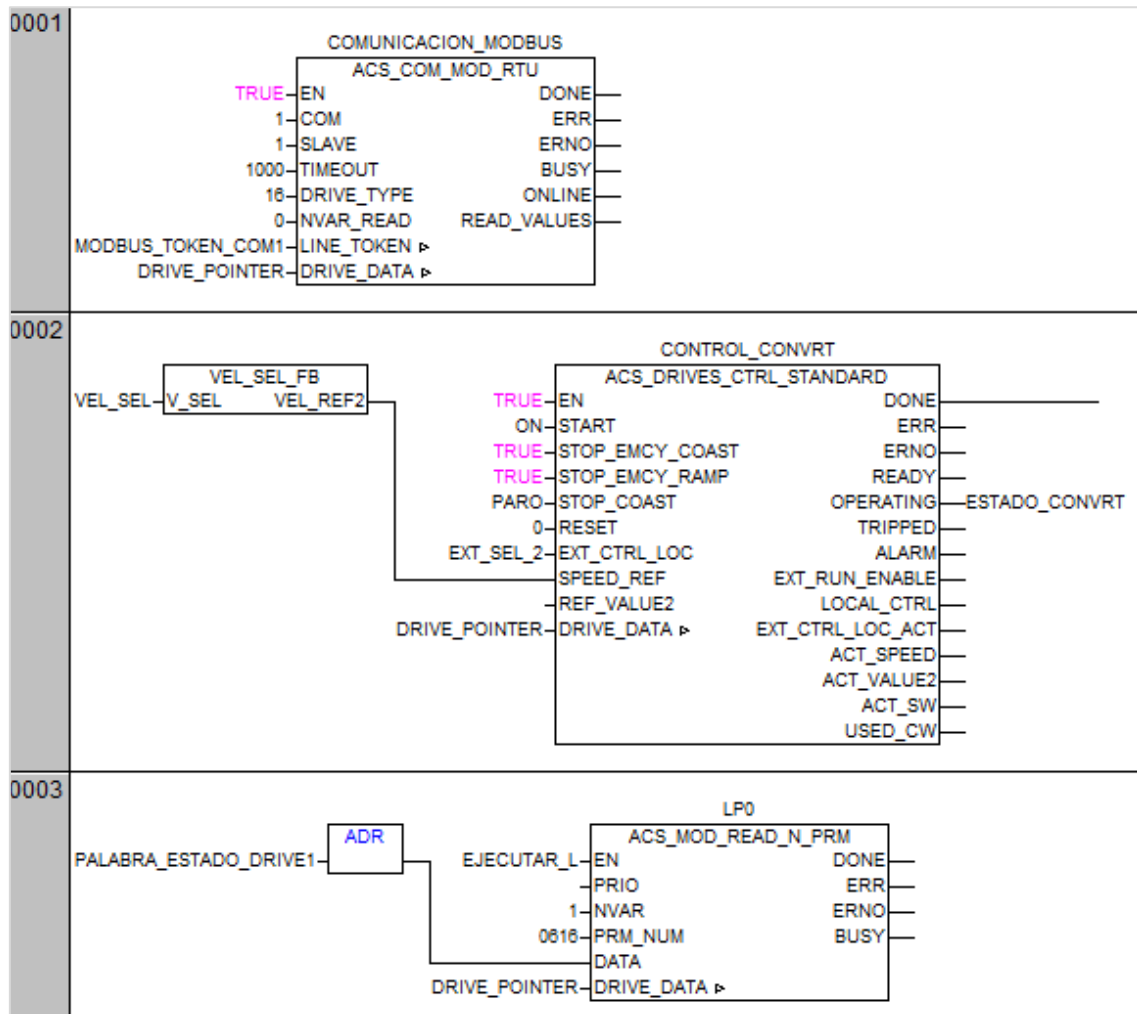
Figura 72: Bloque MOD_WRITE_N_PRM

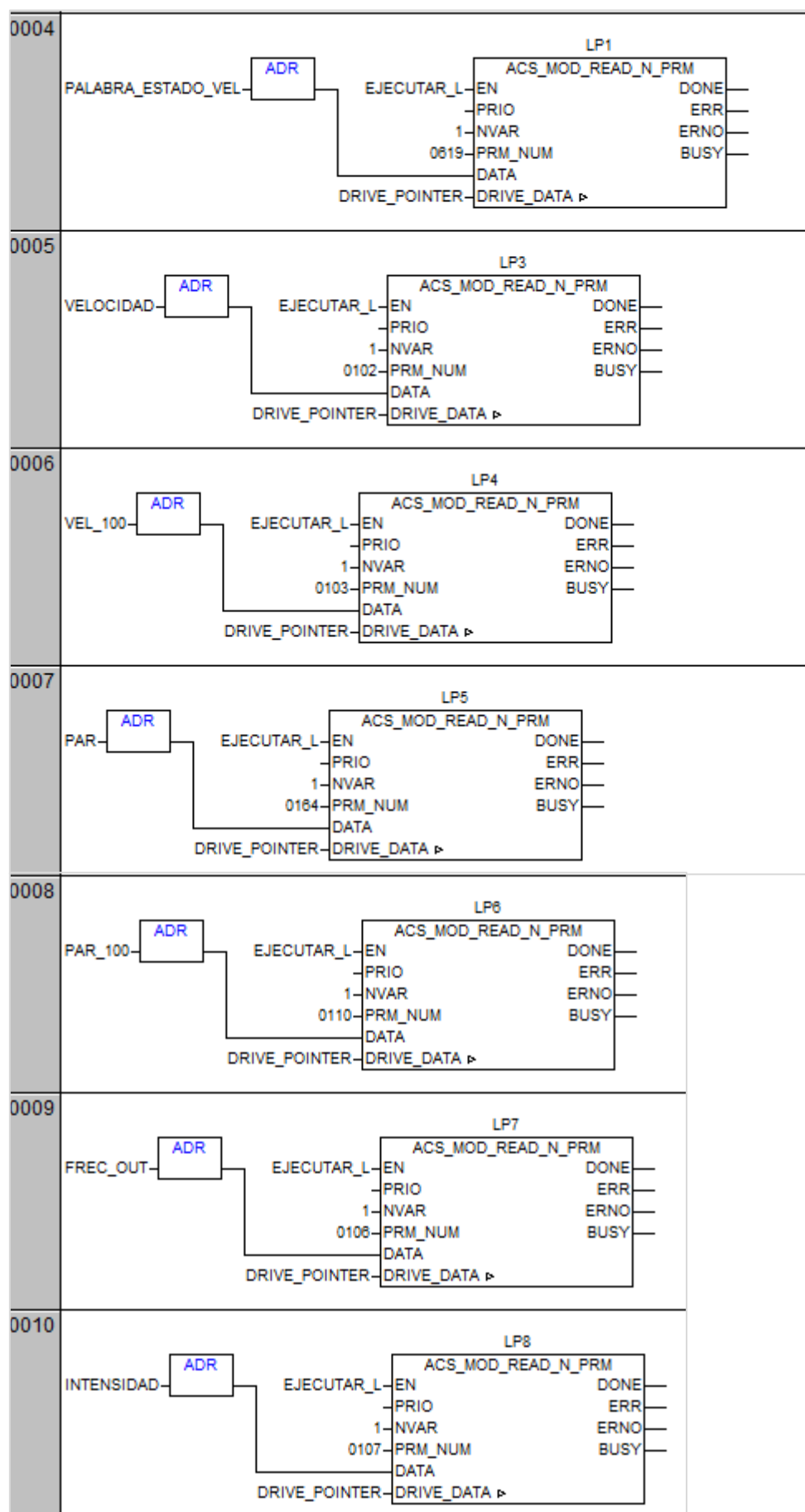
Parámetro	Ajuste seleccionado	Función/Información
EN	EJECUTAR_E (variable de programa)	Habilita (TRUE) /deshabilita (FALSE) el bloque de funciones Tipo de dato: BOOL
PRIO	-	Es un parámetro que no tiene uso por el momento. Está reservado para su uso en el futuro. Tipo de dato: BOOL
NVAR	1	Define el número de variables a leer por el bloque. Tipo de dato: UNIT
PRM_NUM	Depende de la variable que se desee leer del convertidor	Cada variable del convertidor está codificada en un parámetro. La fórmula para codificar cada parámetro es la siguiente: Grupo parám. x 100 + n° parám. Por ejemplo, el parámetro 30.20 se codifica como el parámetro 3020. Tipo de dato: UNIT
DATA	La variable de programa en la que	Es la variable que contiene la información a guardar en el

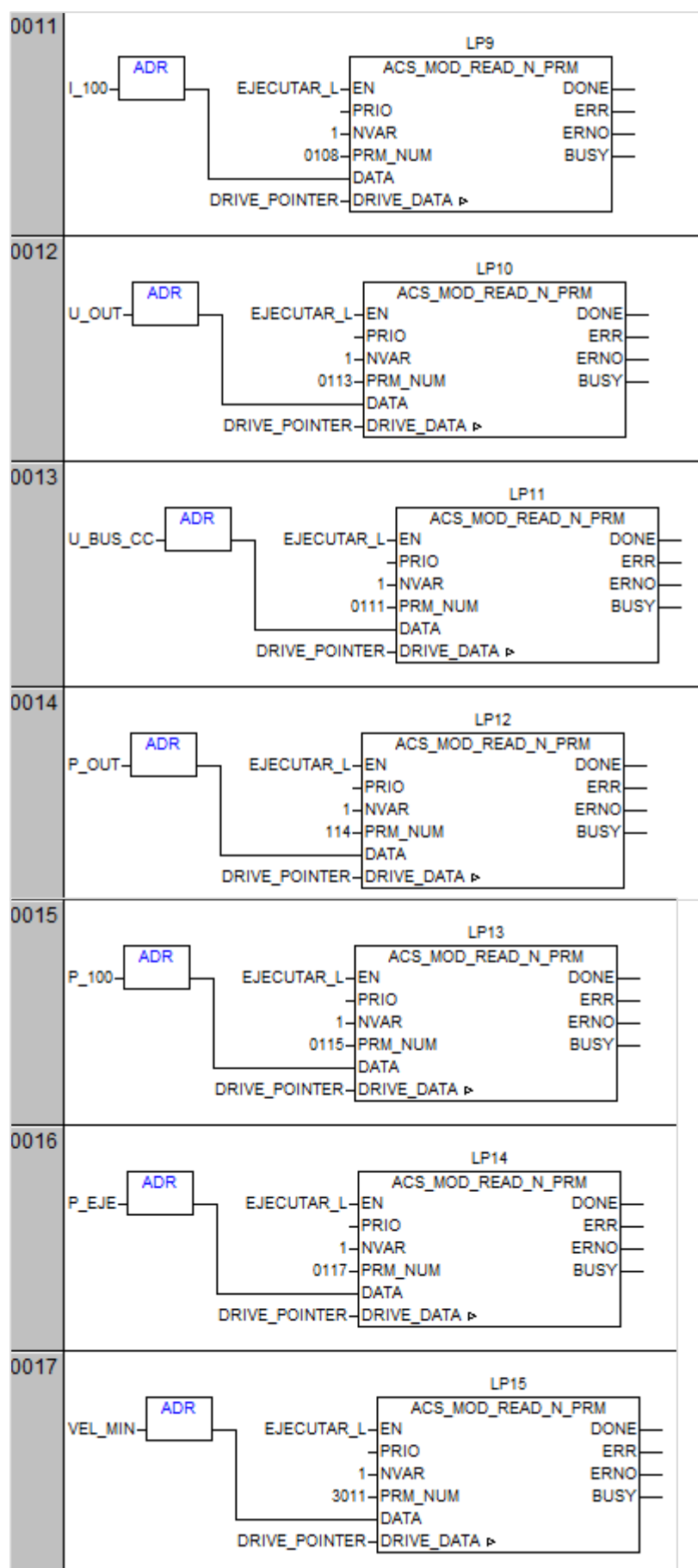
Parámetro	Ajuste seleccionado	Función/Información
	interese guardar la información leída	convertidor. Al tener que estar en formato DWORD, se usa el bloque ADR que es una función de dirección y arroja la dirección de su argumento en DWORD Tipo de dato: DWORD
DRIVE_DATA	DRIVE_POINTER (variable de programa)	Tipo de dato: ACS_DRIVE_DATA_TYPE

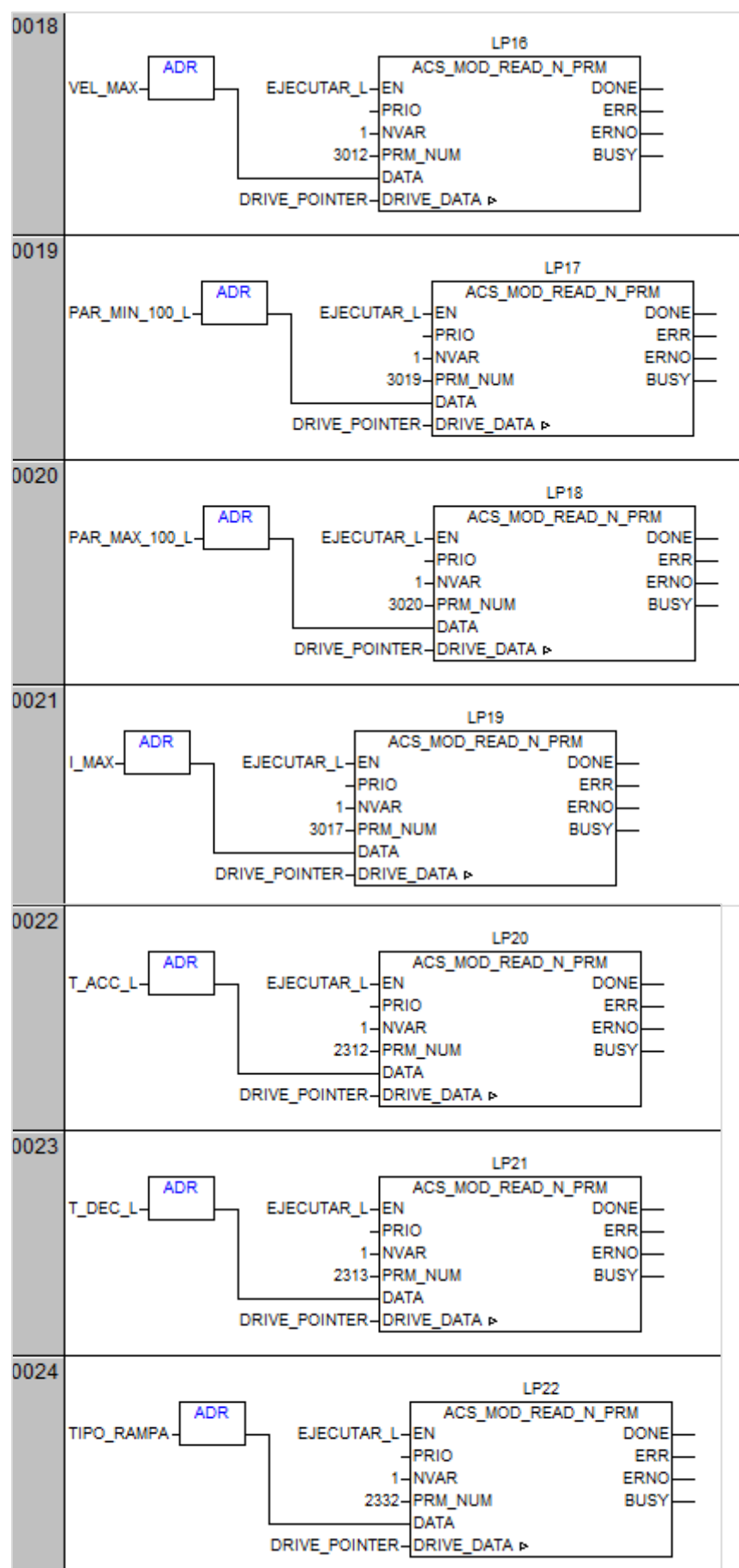
Tabla 22: Entradas ACS_MOD_WRITE_N_PRM

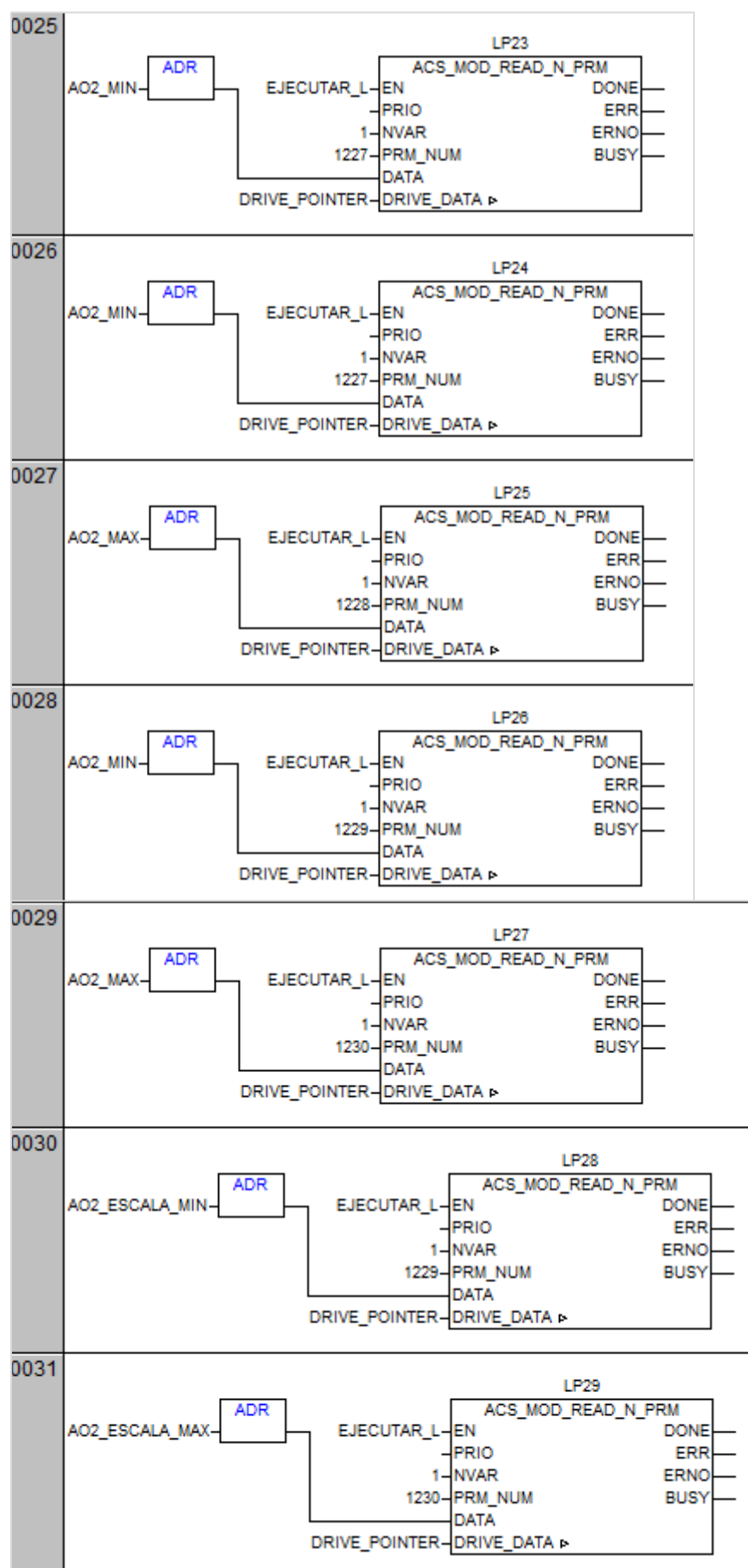
15.2.2.2. Código programa ACS480

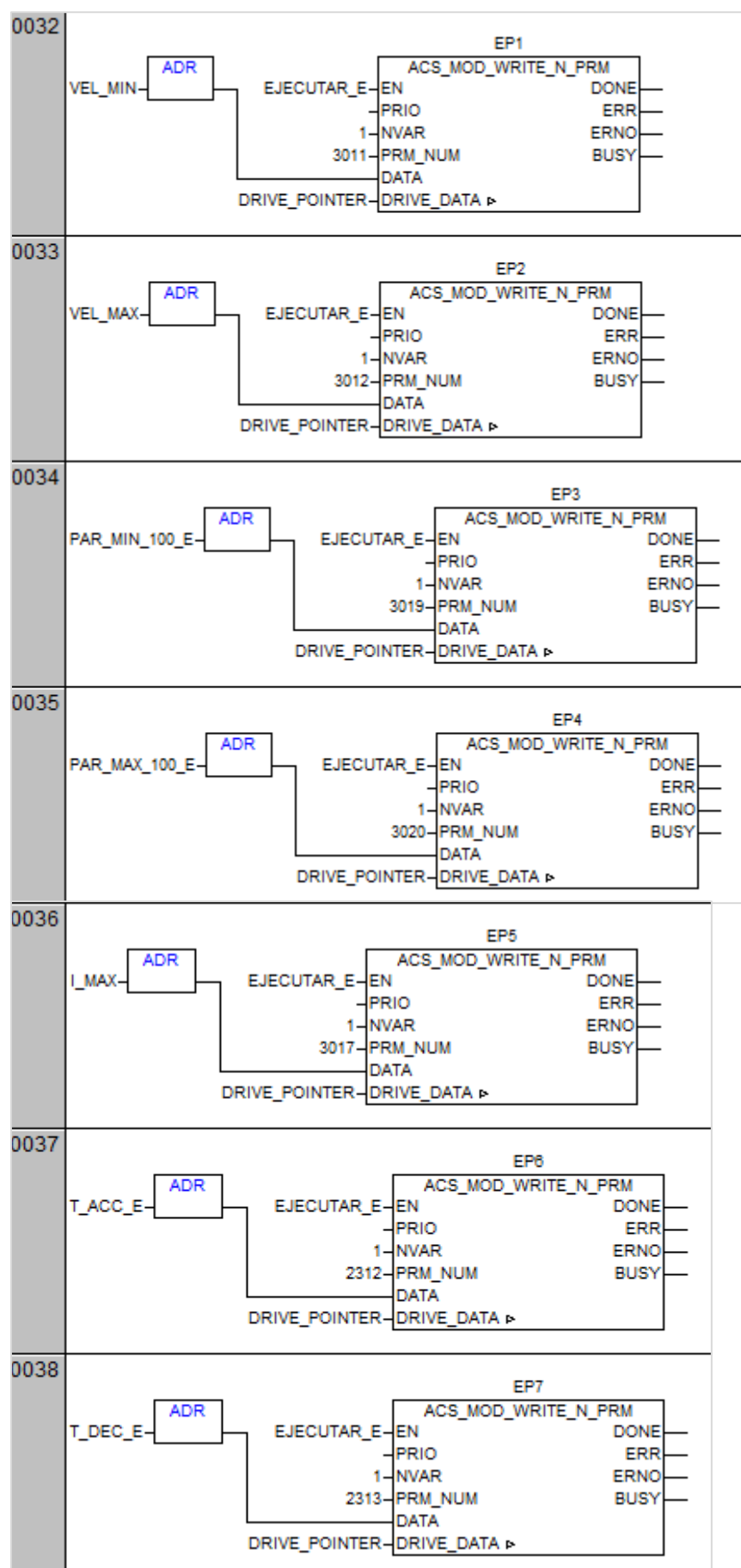












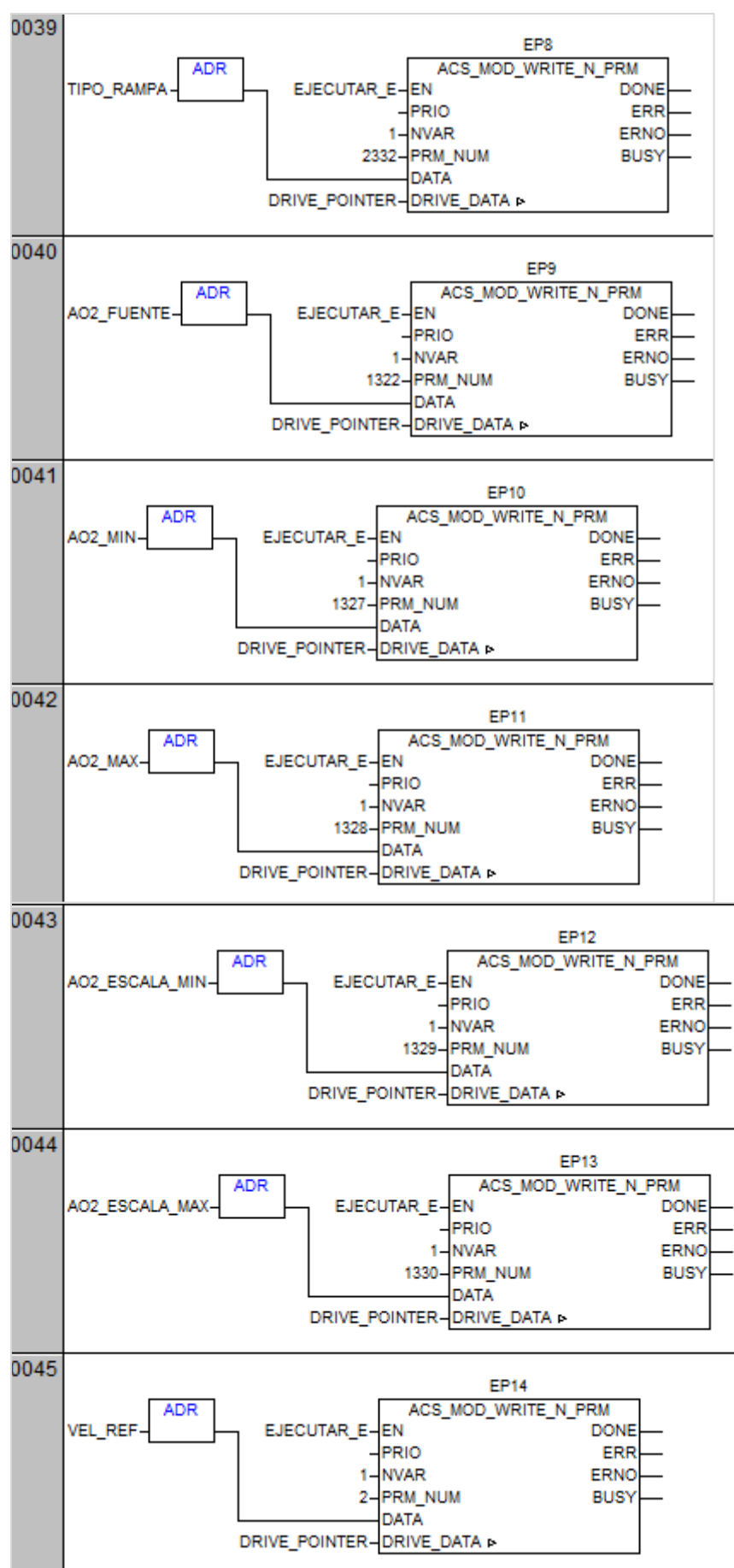


Figura 73: Programa ACS480